Universidad Nacional de Ingeniería
Facultad de Electrotecnia y Computación

TRABAJO MONOGRÁFICO PARA OPTAR AL TÍTULO DE INGENIERO EN COMPUTACIÓN

Sistema web CRM para la empresa Ultimate Business Services y el grupo R Hammond, L.P.

Autores

Br. Lira, Jarbit Josue

Br. López Herrera, Yasser Edsel

Tutor

MSc. Ing. Flor de María Valle Izaguirre

Marzo, 2020

Managua, Nicaragua

Resumen

Este Proyecto de tesis titulado "***Sistema web CRM para la empresa Ultimate Business Services y el grupo R Hammond, L.P***." se enfoca en optimizar los procesos de marketing, ventas y servicio al cliente final, mediante la implementación multi-tenant (literalmente "multi-inquilino" o de tenencia multiple) que permite utilizar una misma base de código, pero áreas de datos separadas para cada empresa cliente.

Este sistema, busca contribuir a una visión unificada de los procesos y servicios brindados por la empresa Ultimate Business Services y el grupo R Hammond, L.P, mediante un único software que suma las capacidades y ventajas de los dos sistemas anteriormente usados. Para esto, se implementa el principio de arquitectura de software multi-tenant, en la cual una sola instancia de la aplicación se ejecuta en el servidor, pero sirviendo a múltiples empresas clientes, logrando un mejor aprovechamiento de los recursos físicos, lógicos y humanos.

Además, CRMX como es llamado internamente el sistema, propone un modelo físico de datos, en el cual los datos están particionados en una única base de datos, con la gran ventaja de que las estructuras de datos están unificadas, sin afectar en nada la confidencialidad de la información.

El sistema CRMX facilita, por lo tanto, el seguimiento continuo de principio a fin a una oportunidad de negocio identificada y gestionada por el funcionario que tiene a cargo determinada cartera de clientes.

El modelo de datos también facilita la personalización de los datos que son particulares y exclusivos a una empresa cliente, permitiendo también la configuración de roles de usuario, de forma totalmente dinámica, sin necesidad de solicitar y esperar una nueva actualización del sistema.

INDICE

1. INTRODUCCIÓN 1

2. OBJETIVOS 2

2.1. OBJETIVO GENERAL 2

2.2. OBJETIVOS ESPECÍFICOS 2

3. JUSTIFICACIÓN 3

4. MARCO TEÓRICO 5

4.1. MODELO DE GESTIÓN DE CLIENTES DEL GRUPO HAMMOND L.P. 5

4.2. ARQUITECTURA DE SOFTWARE MULTI-TENANT 6

4.3. MODELO DE DISTRIBUCIÓN SAAS 7

4.4. FRAMEWORK 8

4.5. WEB SERVICES 8

4.6. HERRAMIENTAS DE DESARROLLO 9

4.6.1. Mockflow 9

4.6.2. C# 10

4.6.3. ASP. NET Web Api 10

4.6.4. HTML 10

4.6.5. Angular 11

4.6.6. Bootstrap 11

4.6.7. Microsoft SQL Server 12

4.6.8. Visual Studio 12

4.6.9. Katalon Studio 12

4.6.10. Amazon Web Services 13

4.6.10.1. Amazon EC2 14

4.6.10.2. Amazon RDS 14

4.6.11. Control de versiones Git 15

4.7. SCRUM COMO METODOLOGÍA DE DESARROLLO 15

4.7.1. El marco de SCRUM 16

4.7.2. Los roles del equipo SCRUM 17

4.7.3. Product Backlog 18
4.7.4. Etapas de SCRUM 18
4.7.4.1. Planeación del Sprint 18
4.7.4.2. Scrum Diario 19
4.7.4.3. Revisión o demostración del sprint 19
4.7.4.4. Retrospectiva del sprint 20
5. ANÁLISIS Y PRESENTACIÓN DE RESULTADOS 21
5.1. METODOLOGÍA 21
5.1.1. Definición del equipo 21
5.1.2. Product Backlog 21
5.1.3. Organización de las etapas de SCRUM 21
5.1.3.1. Planificación del Sprint 21
5.1.3.2. Scrum Diario 22
5.1.3.3. Revisión o demostración del Sprint 22
5.1.3.4. Retrospectiva de cada Sprint 23
5.2. DESARROLLO DEL TEMA 24
5.2.1. Product Backlog Inicial 24
5.2.2. Planificación de los Sprint 28
5.2.3. Diseño del sistema 30
5.2.3.1. Diagrama de componentes 33
5.2.3.2. Mapa de navegación del sitio 34
5.2.3.3. Modelo Conceptual de la Base de Datos CRM y SMA 36
5.2.3.4. Diseño de estructura del sitio web 38
5.2.4. Desarrollo del Sprint #1 43
5.2.4.1. Tareas y sub-tareas desarrolladas 43
5.2.4.2. Base de datos SMA del Sprint 46
5.2.4.3. Base de datos CRM del Sprint 47
5.2.4.4. Casos de Prueba del Sprint 48
5.2.4.5. Retrospectiva de cada Sprint 55
5.2.5. Desarrollo del Sprint #2 55
5.2.5.1. Tareas y sub-tareas desarrolladas 55

5.2.5.2. Base de datos SMA del Sprint 57
5.2.5.3. Base de datos CRM del Sprint 58
5.2.5.4. Casos de Prueba del Sprint 59
5.2.5.5. Revisión del Sprint 64
5.2.5.6. Retrospectiva del Sprint 64
5.2.6. Desarrollo del Sprint #3 65
5.2.6.1. Tareas y sub-tareas desarrolladas 65
5.2.6.2. Base de datos CRM del Sprint 66
5.2.6.3. Casos de Prueba del Sprint 67
5.2.6.4. Revisión del Sprint 74
5.2.6.5. Retrospectiva del Sprint 74
5.2.7. Desarrollo del Sprint #4 75
5.2.7.1. Tareas y sub-tareas desarrolladas 75
5.2.7.2. Base de datos SMA del Sprint 76
5.2.7.3. Base de datos CRM del Sprint 77
5.2.7.4. Casos de Prueba del Sprint 78
5.2.7.5. Revisión del Sprint 84
5.2.7.6. Retrospectiva del Sprint 84
5.2.8. Product Backlog final 84
5.3. IMPLANTACIÓN DEL SISTEMA 90
5.3.1. Servidor Web 90
5.3.2. Base de Datos 94
5.4. CAPACITACIÓN DEL PERSONAL 96
6. CONCLUSIONES 97
7. RECOMENDACIONES 99
8. BIBLIOGRAFÍA 100
ANEXOS 103
COSTOS DEL PROYECTO 104
Software 104
Servidores Amazon Web Services 104

Recursos Humanos *104*
Costos complementarios *105*
Costo total *105*
MODELO FÍSICO DE LA BASE DE DATOS CRM Y SMA 106
DICCIONARIO DE DATOS 108
Sprint #1 ..... *108*
Base de datos del SMA 108
Base de datos del CRM 109
Sprint #2 ..... *116*
Base de datos del SMA 116
Base de datos del CRM 118
Sprint #3 ..... *125*
Base de datos del CRM 125
Sprint #4 ..... *131*
Base de datos del SMA 131
Base de datos del CRM 134
WEB SERVICES 145
MANUAL DE USUARIO
GUÍA DE IMPLANTACIÓN

Índice de Tablas

Tabla 1 Miembros del equipo SCRUM 21
Tabla 2 Product Backlog 27
Tabla 3 Definición Sprint #1..... 28
Tabla 4 Definición Sprint #2..... 29
Tabla 5 Definición Sprint #3..... 29
Tabla 6 Definición Sprint #4..... 30
Tabla 7 Resultado de tareas y sub-tareas del Sprint #1..... 45
Tabla 8 Retrospectiva de Sprint #1 55
Tabla 9 Resultados Sprint #2 57
Tabla 10 Retrospectiva de Sprint #2 64
Tabla 11 Resultados Sprint #3 66
Tabla 12 Retrospectiva de Sprint #3 74
Tabla 13 Resultados Sprint #4 76
Tabla 14 Retrospectiva de Sprint #4 84
Tabla 15 Product Backlog final..... 89
Tabla 16 Calendario de Capacitación..... 96
Tabla 17 Costos de licencias de Software..... 104
Tabla 18 Costo de uso de Amazon Web Services 104
Tabla 19 Salario del jefe de proyectos 105
Tabla 20 Costo de Recursos Humanos..... 105
Tabla 21 Costos complementarios 105
Tabla 22 Costo total del proyecto..... 105
Tabla 23 Diccionario de datos tabla SmaClientUser 108
Tabla 24 Diccionario de datos tabla SmaRefreshToken..... 108
Tabla 25 Diccionario de datos tabla SmaUser..... 109
Tabla 26 Diccionario de datos tabla Account 110
Tabla 27 Diccionario de datos tabla AccountContact 111
Tabla 28 Diccionario de datos tabla Company 111
Tabla 29 Diccionario de datos tabla Contact..... 112

Tabla 30 Diccionario de datos tabla Customer 112
Tabla 31 Diccionario de datos tabla CustomerOpportunity 113
Tabla 32 Diccionario de datos tabla Opportunity 114
Tabla 33 Diccionario de datos tabla OpportunityContact 115
Tabla 34 Diccionario de datos tabla Person 115
Tabla 35 Diccionario de datos tabla SmaApplication 116
Tabla 36 Diccionario de datos tabla SmaClient 117
Tabla 37 Diccionario de datos tabla SmaTenant 117
Tabla 38 Diccionario de datos tabla SmaTenantClient 118
Tabla 39 Diccionario de datos tabla Address 118
Tabla 40 Diccionario de datos tabla AddressPurpose 119
Tabla 41 Diccionario de datos tabla CityDMARegionCode 119
Tabla 42 Diccionario de datos tabla Country 120
Tabla 43 Diccionario de datos tabla County 120
Tabla 44 Diccionario de datos tabla CustomerAddress 121
Tabla 45 Diccionario de datos tabla DMARegionCode 121
Tabla 46 Diccionario de datos tabla file 122
Tabla 47 Diccionario de datos tabla FilePointer 122
Tabla 48 Diccionario de datos tabla OpportunityStage 123
Tabla 49 Diccionario de datos tabla Stage 124
Tabla 50 Diccionario de datos tabla State 124
Tabla 51 Diccionario de datos tabla ZipCode 124
Tabla 52 Diccionario de datos tabla Lead 125
Tabla 53 Diccionario de datos tabla LeadSource 126
Tabla 54 Diccionario de datos tabla OpportunityStatus 126
Tabla 55 Diccionario de datos tabla RecurrentPatternSchedule 128
Tabla 56 Diccionario de datos tabla ScaleType 128
Tabla 57 Diccionario de datos tabla Schedule 129
Tabla 58 Diccionario de datos tabla ScheduleType 129
Tabla 59 Diccionario de datos tabla StatusLead 130
Tabla 60 Diccionario de datos tabla Task 130

Tabla 61 Diccionario de datos tabla TaskStatus 131
Tabla 62 Diccionario de datos tabla SmaAction 131
Tabla 63 Diccionario de datos tabla SmaModule 132
Tabla 64 Diccionario de datos tabla SmaModuleAction 132
Tabla 65 Diccionario de datos tabla SmaRole 133
Tabla 66 Diccionario de datos tabla SmaRoleAction 133
Tabla 67 Diccionario de datos tabla SmaUserAction 134
Tabla 68 Diccionario de datos tabla SmaUserRole 134
Tabla 69 Diccionario de datos tabla ActivityLog 135
Tabla 70 Diccionario de datos tabla ActivityLogObject 135
Tabla 71 Diccionario de datos tabla ActivityType 136
Tabla 72 Diccionario de datos tabla CrmxSettings 136
Tabla 73 Diccionario de datos tabla CustomerCustomFieldValue 137
Tabla 74 Diccionario de datos tabla CustomField 138
Tabla 75 Diccionario de datos tabla CustomFieldValue 138
Tabla 76 Diccionario de datos tabla Department 139
Tabla 77 Diccionario de datos tabla Employee 139
Tabla 78 Diccionario de datos tabla EmployeeSetting 140
Tabla 79 Diccionario de datos tabla EmployeeTerritory 141
Tabla 80 Diccionario de datos tabla EmployeeWorkStation 141
Tabla 81 Diccionario de datos tabla MarketSegment 142
Tabla 82 Diccionario de datos tabla ModuleType 142
Tabla 83 Diccionario de datos tabla StaffEmployee 143
Tabla 84 Diccionario de datos tabla Territory 143
Tabla 85 Diccionario de datos tabla TerritoryZipCode 144
Tabla 86 Diccionario de datos tabla WorkStation 144
Tabla 87 Listado de Web Services del sistema 155

Índice de Ilustraciones

Ilustración 1 Ciclo de vida de Scrum 16
Ilustración 2 Estructura de la aplicación 30
Ilustración 3 Modelo de instancia única compartida 31
Ilustración 4 Arquitectura de la aplicación 32
Ilustración 5 Diagrama de componentes 33
Ilustración 6 Mapa principal del sitio 34
Ilustración 7 Mapa del sitio - Ajustes 35
Ilustración 8 Modelo Conceptual Base de Datos CRM 36
Ilustración 9 Modelo Conceptual de la Base de Datos SMA 37
Ilustración 10 Wireframe - Dashboard 38
Ilustración 11 Wireframe - listar resultados 39
Ilustración 12 Wireframe - Agregar/Editar registros 40
Ilustración 13 Wireframe – Detalle de Cuentas, Oportunidades y Leads 41
Ilustración 14 Wireframe - Ajustes de perfil de usuario 42
Ilustración 15 Wireframe - Ajustes de perfil tenant 42
Ilustración 16 Tablas de datos SMA del Sprint #1 46
Ilustración 17 Tablas de datos del CRM del Sprint #1 47
Ilustración 18 Caso de Prueba BL001 - Ingreso a la Aplicación 48
Ilustración 19 Caso de Prueba BL003 - Agregar Cuentas Clientes 49
Ilustración 20 Caso de prueba BL006 - Crear Contactos 50
Ilustración 21 Caso de prueba BL009 - Agregar Oportunidades 51
Ilustración 22 Caso de prueba BL007 - Filtrar Clientes 52
Ilustración 23 Caso de prueba BL005 - Detalle de Cuenta en Oportunidad 53
Ilustración 24 Caso de Prueba BL004 - Listar Oportunidades por Cuenta 54
Ilustración 25 Tablas de datos SMA del Sprint #2 57
Ilustración 26 Tablas de datos del CRM del Sprint #2 58
Ilustración 27 Caso de Pruebas BL010 - Multiempresas 59
Ilustración 28 Caso de prueba BL013 - Manejar Archivos 60
Ilustración 29 Caso de prueba BL014 - Tareas en cuentas clientes 61

Ilustración 30 Caso de prueba BL016 - Cambiar Propietario de Cuenta 62
Ilustración 31 Caso de prueba BL018 - Cambiar Etapa de Oportunidad 63
Ilustración 32 Tablas de datos del CRM del Sprint #3 66
Ilustración 33 Caso de Prueba BL019 - Actualizar Estado de cliente 67
Ilustración 34 Caso de prueba BL017 - Leads 68
Ilustración 35 Caso de prueba BL024 - Calendario de Citas 69
Ilustración 36 Caso de prueba BL021 - Oportunidad ganada/perdida 70
Ilustración 37 Caso de prueba BL026 - Listar Contactos en Cuentas Cliente 71
Ilustración 38 Caso de prueba BL027 - Listar Tareas en Cuentas Cliente 72
Ilustración 39 Caso de prueba Territorios 73
Ilustración 40 Tablas de datos SMA del Sprint #4 76
Ilustración 41 Tablas de datos del CRM del Sprint #4 77
Ilustración 42 Caso de prueba BL031 - Reportes 78
Ilustración 43 Caso de prueba BL011 - Campos Personalizados 79
Ilustración 44 Caso de Prueba BL022 - Panel de Administración 80
Ilustración 45 Caso de prueba BL029 Roles de Usuario 81
Ilustración 46 Caso de prueba BL033 - Departamentos y áreas laborales 82
Ilustración 47 Caso de prueba BL034 - Listado de Empleados 83
Ilustración 48 Descripción de la instancia de Máquina Virtual corriendo en AWS 90
Ilustración 49 Directorio de Archivos del API 91
Ilustración 50 Directorio de Archivos del SPA 91
Ilustración 51 Directorio de Archivos de FSS 92
Ilustración 52 Directorio de Archivos de SMA 92
Ilustración 53 Configuración de IP de destino en el dominio 93
Ilustración 54 URL de acceso al sistema CRMX 93
Ilustración 55 Detalles del servicio RDS en AWS 94
Ilustración 56 Scripts SQL a ejecutar al momento de crear un nuevo Tenant 95
Ilustración 57 Modelo Físico Base de Datos CRM 106
Ilustración 58 Modelo Físico de la Base de Datos SMA 107

1. Introducción

Ultimate Business Services (UBS), es una empresa fundada hace más de 20 años por Robert Hammond, dueño del grupo R Hammond, L.P; dicho grupo lo conforman empresas como Hammond I, Inc., Great Pines Water Company, Inc., Rnpartners, Inc. y Partsbase, Inc., siendo esta última la que cuenta con un departamento de desarrollo de sistemas web en Nicaragua para el grupo R Hammond, L.P.

Ultimate Business Services es el proveedor líder de suministros como agua potable, café, primeros auxilios, seguridad, conserjería y oficina, en Florida, EUA; actualmente la empresa brinda servicios a más de 9,000 clientes en el área sur de Florida y con la creciente demanda de sus servicios en el resto del Estado, la empresa se ve en la necesidad de gestionar sus clientes actuales y potenciales haciendo uso de dos sistemas CRM (Customer Relationship Management), los cuales poseen distintas características y funcionalidades trabajando en conjunto.

El desarrollo del sistema estará basado en la estructura Multi-Tenant, donde una sola instancia de la aplicación se ejecuta en el servidor, pero con la capacidad de servir a múltiples empresas, la aplicación podrá realizar la partición virtual de sus datos y configuración para que cada empresa tenga una instancia virtual adaptada a sus requerimientos.

El presente trabajo monográfico tiene como finalidad el desarrollo de un sistema de información web CRM para el manejo de la información de los clientes de las empresas del grupo R Hammond, L.P, mejorando los canales de comunicación y soporte, haciendo de la atención al cliente una experiencia personalizada, eficiente y oportuna.

Con la implementación del nuevo sistema, al cual llamaremos CRMX de ahora en adelante, las empresas del grupo R Hammond, L.P. podrán conocer con anticipación los deseos de sus clientes, para brindar con mayor efectividad los productos y servicios que satisfagan sus necesidades.

2. Objetivos

2.1. Objetivo General

Desarrollar un sistema web CRM que permita a la empresa Ultimate Business Services mejorar los procesos de gestión sobre la información de sus clientes.

2.2. Objetivos Específicos

- ✓ Identificar los requerimientos del sistema a través de la exploración de sistemas actualmente en uso y entrevistas sobre los procesos de control de clientes que realiza la empresa.
- ✓ Diseñar un modelo de solución del sistema haciendo uso de herramientas orientadas a objetos.
- ✓ Realizar la codificación de los módulos del modelo seleccionado utilizando la herramienta Visual Studio de acuerdo a la información establecida en el diseño.
- ✓ Realizar pruebas mediante Katalon Studio y ajustes necesarios que aseguren el correcto funcionamiento del sistema en su conjunto.
- ✓ Implantar el sistema web en los servidores Amazon Web Services pertenecientes a la empresa para su debido uso y capacitación de los usuarios finales.

3. Justificación

El proceso de gestión de clientes y ventas de Ultimate Business Services (UBS) se lleva a cabo utilizando dos sistemas informáticos distintos que trabajan en conjunto para sobrellevar la carga de información y requerimientos de la empresa.

Los mayores problemas en el uso de más de un sistema son las posibles inconsistencias y falta de integridad en la información de los clientes, lo cual debilita uno de los eslabones más importantes de la organización, sumándole el tiempo adicional utilizado por los empleados de la empresa para consolidar los datos de los clientes, lo que incrementa los costos operativos de la empresa.

Por ello, se plantea el desarrollo de un nuevo sistema CRM, a fin de optimizar el desempeño de las tareas realizadas por los colaboradores en el manejo de los clientes de la empresa UBS, sumado a la necesidad de los directivos del grupo R Hammond L.P. de integrar a todas las empresas del grupo en un mismo sistema que se adapte completamente a su modelo de gestión de clientes, tomando como fuente de referencia los sistemas que UBS actualmente utiliza.

Para desarrollar un sistema CRM integral es necesario utilizar una arquitectura Multi-tenant, lo que permite el manejo de una sola instancia de la aplicación sirviendo a las empresas del grupo Hammond L.P., lo cual no es posible con los dos sistemas actualmente en uso, por lo tanto, se simplifica el proceso de gestión de clientes sin la complejidad de instalación y esperar actualizaciones o mejoras para cada empresa del grupo.

El principal beneficio del multi-tenant es su rentabilidad. El hecho de compartir software, hardware, desarrollo de aplicaciones y costos de mantenimiento entre los arrendatarios pueden disminuir los costos para cada uno de estos. Asimismo, el hecho de compartir una única instancia de una aplicación puede ofrecer beneficios adicionales, por ejemplo, que todos los clientes del sistema tengan una actualización simultánea cada vez que se actualice la aplicación. (Goldszmidt, 2011)

El sistema contará con las características básicas de un CRM, tomando en cuenta las virtudes de los sistemas utilizados actualmente y mejoras solicitadas por la empresa Ultimate Business Services y el grupo R Hammond, L.P; lo que resultará en una solución amigable y adaptada a las necesidades del cliente.

El sistema web propuesto permitirá además, acceso desde smartphones, computadoras o tablets, lo que garantizará que el proceso de gestión de prospectos, oportunidades y cuentas, pueda realizarse fácil y eficientemente, en el momento oportuno, desde diferentes ubicaciones con acceso a internet.

4. Marco teórico

El Marco teórico brinda las referencias más relevantes para entender los principios, tecnologías y metodologías utilizadas en el desarrollo del trabajo monográfico.

Primero se describe el modelo de gestión de clientes, es decir, la forma en que el grupo Hammond L.P asume la responsabilidad de atender y asesorar al cliente para asegurar que éste forme parte de su cartera de clientes.

Posteriormente, se describe la arquitectura de software que será utilizada para el modelado del sistema Multi-Tenant, el modelo de distribución basado en software como un servicio y luego se presenta la definición de framework, que es el conjunto de herramientas que permiten un desarrollo ágil.

A continuación, se presenta un listado y una breve descripción de las herramientas de software necesarias para el cumplimento de los objetivos de este trabajo monográfico, principalmente para las fases de desarrollo, pruebas e implantación.

Finalmente, se describe la metodología de desarrollo SCRUM utilizada como marco de trabajo ágil para desarrollar y entregar incrementos del producto software en periodos de tiempo cortos, aprovechando la colaboración y trabajo en equipo.

4.1. Modelo de gestión de clientes del grupo Hammond L.P.

CRM es la sigla utilizada para Customer Relationship Management y se refiere al conjunto de prácticas, estrategias de negocio y tecnologías enfocadas en la relación con el cliente. (Salesforce, 2018)

El grupo R Hammond L.P. maneja tres eslabones importantes dentro de su modelo de gestión de clientes, los clientes potenciales o **Prospectos**, los clientes que reciben una oferta económica llamada **Oportunidad** y los clientes que ya

forman parte de su cartera conocidos como **Cuentas**.

Un ***Prospecto*** (Leads en inglés), es toda aquella empresa o persona natural que podría estar interesada en los productos que comercializa una empresa, el Prospecto puede ser adquirido por distintas vías de contacto, por ejemplo, una llamada telefónica, una campaña de marketing o bien, haber llegado por medio de una recomendación, pero también la empresa podrá convertir en Prospecto a aquellas personas o empresas que considere estén interesados en sus productos.

Una ***Oportunidad*** (Opportunity en inglés), es una oferta de negocio presentada a un cliente, esta oferta puede generarse desde los siguientes escenarios:

1. Convertir a un Prospecto en Oportunidad al presentar una oferta de servicios.
2. La Oportunidad se crea directamente cuando una persona o empresa ha solicitado una oferta de servicios sin ser un Prospecto.
3. Se presenta una oferta de servicios a una cuenta que ya posee negocios con la empresa, convirtiéndose así en una nueva Oportunidad con un cliente actual.

La oportunidad está compuesta por diversos estados que permiten medir el avance de las negociaciones con el cliente.

El cliente decide si aceptar o no la oferta de servicios, lo que se traduce en Oportunidad Ganada o Perdida, la Oportunidad Ganada pasa a ser una nueva ***Cuenta*** (Account en inglés), a partir de este momento la empresa o persona natural oficialmente está haciendo negocios con la empresa oferente.

4.2. Arquitectura de software Multi-Tenant

Se refiere a un principio de arquitectura de software, donde un sistema comparte varios clientes, es decir que trabaja sobre una misma instancia de software, la cual se ejecuta desde un servidor, a este tipo de arquitectura se la llama

arquitectura multi propietario o multiusuario. (Bustamante Granda, 2017)

Según el artículo de INNOVA, otra forma de conceptualizar Multi–Tenancy se refiere a la habilidad para ubicar varios clientes dentro de una única instancia del software que se ejecuta en un servidor.

En un entorno Multi-Tenant, todas las empresas consumen el servicio desde la misma plataforma tecnológica, desde el modelo de datos, servidores y las capas de base de datos.

Para lograr una distribución de aplicaciones Multi-Tenant hay que centrarse en tres aspectos fundamentales:

1. El mecanismo para la distribución de recursos
2. Aislamiento de seguridad
3. El mecanismo de personalización

Pilares de la arquitectura Multi-Tenancy

- Disponibilidad
- Separación segura
- Garantía de servicios
- Capacidad de gestión

4.3. Modelo de distribución SaaS

SaaS – Software as a Service es un modelo de distribución de software donde el software y los datos se alojan en servidores del proveedor y se accede con un navegador web a través de Internet.

El proveedor da el servicio de mantenimiento, operación diaria, y soporte del software usado por el cliente, y el cliente paga por el uso dado a la aplicación.

El Software SaaS tiene una característica que hace que el modelo sea especialmente eficiente: es multitenancy. Esta es la propiedad que permite ofrecer la misma aplicación a muchos usuarios y así distribuir el coste de la infraestructura y del mantenimiento entre todos. Es decir, una aplicación que permite con una sola instancia y una sola base de datos dar servicio a todos los clientes. (Portal SaaS, s.f.)

4.4. Framework

En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. (Gutiérrez, 2005)

Según Orix System (2015) los mayores beneficios de los frameworks son:

- Evitar escribir código repetitivo
- Utilizar buenas prácticas
- Permitir hacer cosas avanzadas
- Desarrollar más rápido
- Soporte de comunidad
- Reutilización
- Seguridad

4.5. Web Services

Un web service es una aplicación que puede ser descrita, publicada, localizada e invocada a través de una red, generalmente Internet. Combinan los mejores

aspectos del desarrollo basado en componentes y la Web.

Al igual que los componentes, un web service es un conjunto de funcionalidades que se encuentran dentro de una caja negra, que pueden ser reutilizados sin preocuparse de cómo fueron implementados. A diferencia de la actual tecnología de componentes, no son accedidos por medio de protocolos específicos del modelo de objetos como son RMI, DCOM o IIOP; sino que son accedidos utilizando protocolos web como pueden ser HTTP y XML.

La interface del web service está definida en términos de los mensajes que el mismo acepta y retorna, por lo cual los consumidores del web service pueden ser implementados en cualquier plataforma y en cualquier lenguaje de programación, solo tiene que poder crear y consumir los mensajes definidos por la interfaz del web services. (GxTechnical, s.f.)

4.6. Herramientas de desarrollo

4.6.1. Mockflow

Mockflow es una solución de software basada en la nube diseñada para la creación de *Wireframe*, es una solución orientada a desarrolladores de sitios web y software, diseñadores, profesionales de UX y comercializadores. Les permite realizar tareas, diseñar prototipos de interfaz de usuario para sitios web y aplicaciones móviles, y colaborar con proyectos en tiempo real. (Crozdesk TM, s.f.)

Un Wireframe es un esquema visual que transmite un nivel básico de comunicación, estructura y comportamiento durante el diseño de un sistema. Los wireframes son diseños de baja fidelidad que evitan la inclusión de una interfaz de usuario detallada o un diseño visual, transmitiendo lo suficiente para mostrar la idea central. (Every Interaction., s.f.)

4.6.2. C#

C# es un lenguaje elegante y seguro orientado a objetos que permite a los desarrolladores crear una variedad de aplicaciones seguras y robustas que se ejecutan en .NET Framework, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes. (Microsoft Docs, 2015).

4.6.3. ASP. NET Web Api

ASP.NET Web API es un framework para construir APIs (Application Programming Interfaces), construida sobre el .NET Framework. Es una especie de librería más liviana y simple que WCF (Windows Communication Foundation) orientada completamente al protocolo HTTP, que permite construir servicios web sin estar sujetos a un estilo o arquitectura definido (por ejemplo, REST o RPC). (Souvenirs de IT, s.f.)

Web API utiliza el URL de las peticiones y obtiene resultados en formato JSON. JSON es un acrónimo de JavaScript Object Notation y es un formato ligero para intercambio de datos. Para las personas resulta de fácil lectura y escritura mientras que para las máquinas les resulta fácil analizarlo y generarlo. (Muñoz Serafín, 2015)

4.6.4. HTML

HTML significa "Lenguaje de Marcado de Hipertexto" por sus siglas en inglés

"HyperText Markup Language", es un lenguaje que pertenece a la familia de los "lenguajes de marcado" y es utilizado para la elaboración de páginas web. El estándar HTML lo define la W3C (World Wide Web Consortium) y actualmente HTML se encuentra en su versión HTML5.

Cabe destacar que HTML no es un lenguaje de programación ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de los lenguajes de programación, por lo que HTML genera únicamente páginas web estáticas, sin embargo, HTML se puede usar en conjunto con diversos lenguajes de programación para la creación de páginas web dinámicas. (Acerca de HTML, s.f.)

4.6.5. Angular

Angular es un framework que se emplea para todo tipo de creaciones y ediciones de aplicaciones y software que se realiza destinado a aplicaciones web y a Javascript. A lo largo de los años, este framework ha ido evolucionando y creciendo hasta posicionarse de la mejor manera posible y contar con una mayor utilidad para el mercado y para los desarrolladores profesionales.

La nueva versión de Angular se llama AngularJS y logra mucho más, a la vez que mejora lo ya existente. Permite crear aplicaciones y desarrollar proyectos tanto para ordenadores y para sistemas operativos de escritorio como para móvil, tablet y otro tipo de dispositivos, lo que le da una utilidad mayor y la hace más recomendable, aconsejable y altamente funcional para los desarrolladores. (Forma Talet - Bussines School, s.f.)

4.6.6. Bootstrap

Bootstrap es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño

adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien. Es Open Source o código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones.

4.6.7. Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de base de datos relacional desarrollado por Microsoft. Como un servidor de base de datos, es un producto de software con la función principal de almacenar y recuperar datos según lo solicitado por otras aplicaciones de software que pueden funcionar ya sea en el mismo ordenador o en otro ordenador a través de una red (incluyendo el Internet). (Biblioteca Profesional, 2016)

4.6.8. Visual Studio

Microsoft Visual Studio. Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles. (EcuRed, 2010)

4.6.9. Katalon Studio

Katalon Studio es una poderosa solución de automatización que ayuda a abordar el desafío de carecer de la integración entre los frameworks open-source de

automatización de pruebas en medio de soluciones comerciales inalcanzables.

Ofrece un conjunto completo e integrado de funciones que van desde la grabación de pasos de prueba, generación de scripts hasta la ejecución y generación de informes de resultados de pruebas para la automatización de pruebas de aplicaciones web y móviles (Yang, 2017)

Según Yang (2017) Katalon Studio revoluciona el uso de marcos de automatización de pruebas de código abierto como Selenium y Appium al eliminar sus complejidades técnicas para permitir a los probadores configurar, crear, ejecutar, informar y gestionar de manera eficiente sus pruebas automatizadas.

También ofrece una alternativa viable a las soluciones de automatización de pruebas comerciales que son inaccesibles para muchos equipos pequeños y medianos.

Katalon Studio en primera instancia graba los pasos que sigue el usuario capturando los nombres, ID y clases de los objetos donde se efectúa un clic, se ingresa un texto o selecciona un ítem de una lista con la finalidad de al momento de realizar una prueba automatizada se recorren el mismo flujo de proceso que el usuario realizó, en la configuración de la prueba se puede cambiar el detalle de datos a ingresar en las cajas de texto o selecciones de las listas.

4.6.10. Amazon Web Services

Amazon Web Services (AWS) es el proveedor Cloud público pionero y líder por estar al servicio de más de un millón de clientes activos en más de 190 países, por incluir numerosos servicios para el despliegue de aplicaciones, por disponer de diferentes regiones (datacenters en diferentes localizaciones geográficas) (Morales, 2016)

4.6.10.1. Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores.

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener e iniciar nuevas instancias de servidor en cuestión de minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, en función de sus necesidades. Amazon EC2 cambia el modelo económico de la informática al permitir pagar solo por la capacidad que utiliza realmente. Amazon EC2 les brinda a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarlas de los casos de error comunes. (AWS Amazon, 2019)

4.6.10.2. Amazon RDS

Con Amazon Relational Database Service (Amazon RDS), es sencillo configurar, utilizar y escalar una base de datos relacional en la nube. El servicio suministra capacidad rentable y escalable al mismo tiempo que automatiza las arduas tareas administrativas, como el aprovisionamiento de hardware, la configuración de bases de datos, la implementación de parches y la creación de copias de seguridad. Lo libera de estas tareas para que pueda concentrarse en sus aplicaciones y darles el rendimiento rápido, la alta disponibilidad, la seguridad y la compatibilidad que necesitan.

Amazon RDS está disponible para varios tipos de instancias de base de datos (optimizadas para memoria, rendimiento u operaciones de E/S) y le proporciona seis motores de bases de datos conocidos entre los que elegir, incluidos Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database y SQL Server. Puede

usar AWS Database Migration Service para migrar o replicar sus bases de datos existentes en Amazon RDS con facilidad. (Amazon AWS, 2019)

4.6.11. Control de versiones Git

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. (Straud & Chaco, 2014)

Entre sus principales características están:

1. Un sistema de control de versiones distribuido.
2. No depende de acceso a la red o un repositorio central.
3. Enfocado a la velocidad, uso práctico y manejo de proyectos grandes
4. Es extremadamente rápido.
5. Autenticación criptográfica del historial.
6. Formato de archivo muy sencillo y compacto.
7. 100% distribuido

El control de versiones de este proyecto se realizará utilizando la herramienta **Bitbucket,** esta herramienta permite crear repositorios privados, limitando así el acceso a usuarios debidamente autorizados al código fuente.

4.7. SCRUM como metodología de desarrollo

Existen diversos métodos para el proceso de desarrollo de software, cada uno de ellos nos brindan una serie de herramientas, paso y técnicas para tener éxito a lo largo del ciclo de vida del software.

Scrum es un proceso de administración y control que atraviesa la complejidad para enfocarse en la construcción de productos que satisfagan las necesidades del negocio. La administración y los equipos pueden poner sus manos sobre los

requisitos y las tecnologías, nunca soltar, y entregar productos en funcionamiento, de forma incremental y empírica. (Schwaber K., Sutherland J.; 2012)

Scrum en sí mismo es un marco simple para una efectiva colaboración de equipo en productos complejos.

4.7.1. El marco de SCRUM

Scrum se ha convertido en el marco ágil de trabajo más usado en el desarrollo de software en el mundo. Uno de los beneficios de Scrum es la entrega de incrementos de producto en periodos de tiempo cortos. El trabajo se basa en la colaboración y trabajo en equipo. Algunos aspectos como el descubrimiento, el empirismo y la inspección continua son claves para lograr un proceso de mejora continua y de innovación en el producto a construir. Scrum está basado en un proceso de control empírico y sirve para construir productos complejos de forma sostenida. (Francia, 2017)

La ilustración 1 representa a Scrum en acción, tal como lo describe el libro Software in 30 Days (Schwaber & Sutherland, 2012) que nos lleva desde la planificación hasta la entrega del software.

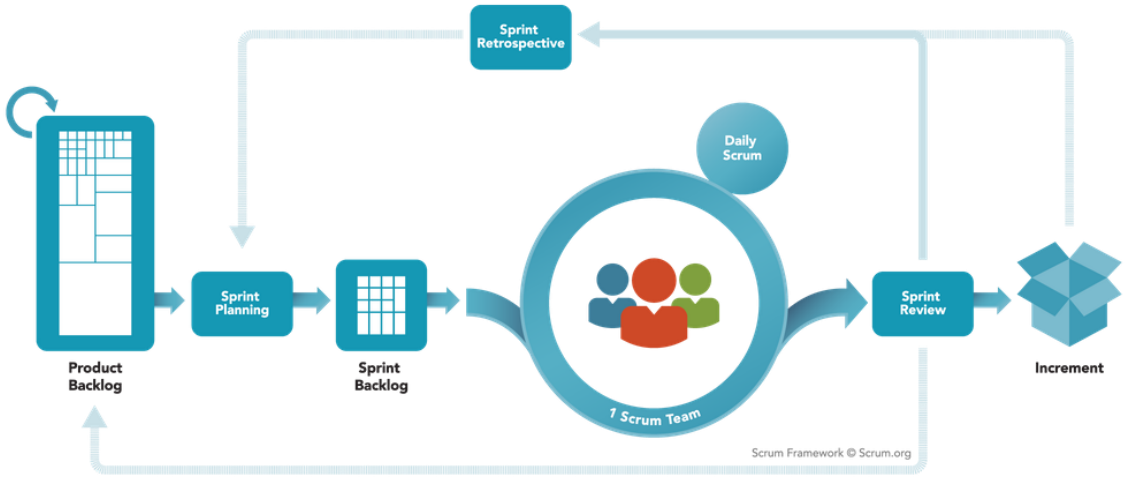

Ilustración 1 Ciclo de vida de Scrum

4.7.2. Los roles del equipo SCRUM

El Scrum Team consta de un Product Owner, el Development Team y un Scrum Master. Los Scrum Teams se auto organizan y tienen funciones cruzadas. Los equipos auto-organizados eligen la mejor manera de llevar a cabo su trabajo, en lugar de ser dirigidos por otros fuera del equipo. Los equipos multifuncionales tienen todas las competencias necesarias para realizar el trabajo sin depender de otros que no formen parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad en el equipo. (Gonzales, 2017)

Scrum Master: Se encargará de conducir al resto del equipo por la metodología SCRUM asegurándose de que todos los procesos de SCRUM sean implementados correctamente; tiene el deber de ayudar al Product Owner a mejorar la comunicación e información de cada evento relacionado al proyecto, también trata de eliminar todo aquello que pueda provocar un retraso en cada avance y generar un seguimiento e informes del progreso de proyecto.

Product Owner: Es el encargado de velar por los intereses del cliente, en algunos casos es la misma persona, tiene la tarea otorgar al equipo de trabajo una lista priorizada de elementos (también conocida como backlogs) que el cliente espera del proyecto la cual se convierte en el principal recurso de planificación en SCRUM; también debe de asegurarse de que cada tarea que conforma el backlog sea fácil de entender para cada uno de los participantes del proyecto. Debe de participar en la definición de cada Sprint y revisar el producto al término de cada uno para determinar si se cumplió con lo deseado.

Development Team: Se encargan de crear cada uno de los artículos pertenecientes al backlog siempre y cuando se cumpla la planificación de cada sprint, deben de tener la capacidad de ser auto-organizados y multifuncionales para cumplir con cada una de las tareas que se les asigna.

4.7.3. Product Backlog

El Backlog no es más que una lista de todo lo que debe hacerse para convertir la visión en realidad. Esta lista existe y evoluciona a lo largo del proceso, es el mapa o la hoja de ruta del producto.

En cualquier momento del proyecto, la lista de objetivos pendientes es la única y definitiva vista panorámica de todo lo que el equipo podría hacer, por orden de prioridades. Existe una sola lista de objetivos pendientes. Esto quiere decir que el responsable de producto tiene que tomar decisiones sobre las prioridades del proceso. (ProAgilist, 2016)

Debería consultar con todos los interesados y con el equipo para asegurarse de que representan tanto lo que quiere el cliente como lo que es factible construir.

Durante el desarrollo de cada Sprint generalmente surgen detalles adicionales, en el diseño, en la funcionalidad, la verificación de actividades, etc., creando así nuevos ítems en el Product Backlog.

4.7.4. Etapas de SCRUM

4.7.4.1. Planeación del Sprint

El equipo, el Scrum Master y el responsable de producto se sientan a planificar el sprint.

Habitualmente, casi todo el mundo hace Sprints de una o dos semanas procurando no exceder un tiempo máximo a cuatro semanas, el equipo mira el principio de la lista de objetivos pendientes y hace una previsión de cuánto pueden tener terminado en este sprint. Si el equipo ya ha hecho algún sprint, deberían tener en cuenta los puntos que hicieron en el último. También es el momento para que el equipo y el responsable de producto se aseguren de que todo el mundo entiende exactamente cómo estos ítems van a lograr crear la visión. Además, en esta reunión todos deben ponerse de acuerdo en la meta, lo que todos quieren lograr en ese sprint.

Uno de los pilares del Scrum es que una vez que el equipo se ha comprometido con lo que creen que pueden terminar en un sprint, no hay vuelta atrás. No se puede cambiar y no se le puede añadir nada. El equipo tiene que ser capaz de trabajar de forma autónoma durante todo el sprint, para terminar lo que previeron que podían hacer. (ProAgilist, 2016)

4.7.4.2. Scrum Diario

Éste es el pulso vital del Scrum. Cada día a la misma hora, durante no más de quince minutos, el equipo y el Scrum Master se ven y responden a tres preguntas:

¿Qué hiciste ayer para ayudar al equipo a terminar el sprint?

¿Qué vas a hacer mañana para ayudar al equipo a terminar el sprint?

¿Qué obstáculos se interponen en tu camino o el del equipo?

Ya está. En eso consiste la reunión. Si dura más de quince minutos usted está haciendo algo mal. Esto ayuda al equipo a saber exactamente en qué punto está cada ítem del sprint. (ProAgilist, 2016)

4.7.4.3. Revisión o demostración del sprint

Ésta es la reunión en la que el equipo muestra lo que ha construido durante el sprint. Puede estar presente cualquiera, no sólo el responsable de producto, el Scrum Master y el equipo, sino los directivos de la empresa, los jefes, los clientes, todo el que quiera. Es una reunión abierta en la que el equipo explica lo que han podido pasar a la columna de **“hecho”** durante el sprint.

El equipo debería mostrar únicamente lo que se ajuste perfectamente a la definición de **“hecho”**. Aquello que esté completamente terminado y que se puede entregar porque no necesita más trabajo. Puede no ser un producto terminado, pero debería ser una característica del mismo, que está lista para empezar a funcionar. (ProAgilist, 2016)

4.7.4.4. Retrospectiva del sprint

Después de que un equipo haya mostrado lo que ha conseguido durante el último sprint se sientan a reflexionar sobre lo que ha ido bien, lo que podría hacerse mejor y lo que se podría perfeccionar en el siguiente sprint. ¿Qué mejora puede incorporar el equipo al proceso de forma inmediata?

No estamos buscando a quién echarle la culpa; estamos analizando el proceso. ¿Por qué eso fue así? ¿Por qué se nos escapó aquello? ¿Qué podría hacernos ser más rápidos? Es crucial que la gente asuma la responsabilidad de su proceso y resultados y trate de encontrar soluciones como equipo. A su vez, las personas tienen que tener la valentía de plantear los problemas con los que realmente se están encontrando de una forma constructiva. Para solucionar y no a acusar. El resto del equipo debe tener la madurez de escuchar esa opinión, tenerla en cuenta y buscar una solución, en lugar de ponerse a la defensiva.

Al final de la reunión, el equipo y el Scrum Master deberían haberse puesto de acuerdo en una mejora del proceso que incorporarán en el siguiente sprint. (ProAgilist, 2016)

5. Análisis y presentación de resultados

5.1. Metodología

A continuación, se describe cada uno de los pasos de SCRUM que han sido aplicados para la realización de este proyecto.

5.1.1. Definición del equipo

Definición de roles del proyecto	
Scrum Master	Ing. Rodrigo García
Product Owner	Lic. Samoel Garrido
Development Team	Jarbit Josue Lira Yasser Edsel López Herrera

Tabla 1 Miembros del equipo SCRUM

5.1.2. Product Backlog

La lista de tareas ha sido levantada por el Product Owner y contiene las necesidades del cliente, con el fin de poder crear y utilizar adecuadamente el producto deseado.

Para el levantamiento del Product Backlog el Product Owner ha realizado un conjunto de preguntas cerradas a personal debidamente seleccionado que contaba con la experiencia de uso de los dos sistemas que se plantearon reemplazar.

5.1.3. Organización de las etapas de SCRUM

A continuación, se describen los acuerdos del equipo concernientes a cada una de las etapas de SCRUM cuyo objetivo final es la entrega del sistema web CRMX.

5.1.3.1. Planificación del Sprint

Antes de iniciar cada Sprint el equipo SCRUM se reunirá para determinar cómo se desarrollara y que resultados se esperan al finalizar cada iteración; una vez analizado el Product Backlog y haber sido comprendido por cada uno de los

miembros se decidirá cuales ítems declarados de prioridad “Alta” por el Product Owner serán abordados en el próximo Sprint, también para equilibrar el Sprint se seleccionarán ítems con prioridades bajas y medias donde se considere que requieren poco esfuerzo de desarrollo.

Junto con la selección de cada ítem que formara parte del Sprint el equipo debe determinar cuándo iniciara y terminara el Sprint enfocados en no prolongar el mismo por un lapso de tiempo mayor a un mes, si en la planificación del Sprint se calcula un tiempo mayor a un mes lo mejor es retirar ítems de la lista y programarlos para el siguiente Sprint; si ya se ha realizado un Sprint del proyecto con anterioridad el equipo contara con una excelente referencia de tiempo estimado de desarrollo y podrá apuntar a mejorar los tiempos de entrega de cada Sprint subsiguiente.

5.1.3.2. Scrum Diario

Las reuniones diarias están programadas para efectuarse todos los días a primera hora de la jornada laboral por un lapso no mayor a 15 minutos con la finalidad de enfocarse únicamente en los análisis de avances efectuados el día anterior, las tareas del día de reunión, obstáculos que se interponen al equipo y previsiones para el día siguiente.

Es tarea del Scrum Master ayudar al equipo a superar cualquier obstáculo que se interponga entre ellos y los tiempos de entrega, también consultar a cada miembro si está en capacidad de apoyar a cualquier otro miembro que se encuentre en dificultades.

5.1.3.3. Revisión o demostración del Sprint

Al término de cada Sprint se realiza una reunión del equipo SCRUM donde también pueden participar otros interesados del proyecto; el objetivo de esta reunión es mostrar y explicar los resultados del Sprint. Como lo dicta la metodología solo se mostrarán aquellos ítems cuyo estado se encuentre en **“Hecho”**; solo se pueden entregar aquellos ítems que no requieran más trabajo y

aunque el producto total no esté terminado se consideren una característica lista para operar.

5.1.3.4. Retrospectiva de cada Sprint

Luego de haber demostrado los resultados del Sprint, el equipo se reunirá para realizar un análisis de lo que se ha hecho bien, lo que se podría hacerse mejor y lo que se podría perfeccionar para el siguiente Sprint, de esta reunión no podrán participar personas ajenas al equipo.

Uno de los objetivos de este proceso no es encontrar culpables ni afectar la productividad del equipo, solo enfocarse en indagar en que se está fallando y cómo se puede mejorar como equipo. Estas reuniones se pueden programar inmediatamente después de la entrega del Sprint.

5.2. Desarrollo del tema

5.2.1. Product Backlog Inicial

La elaboración de Product Backlog inicial se facilitó debido a la participación directa Product Owner, quien cuenta con los conocimientos informáticos y del modelo del negocio, sumado a la comunicación directa que tenía con los usuarios principales, lo que disminuyó la participación de los tesistas en el proceso de entrevistas a los usuarios finales.

A continuación, en la Tabla 2, se muestra el Product Backlog inicial que contiene las tareas que se deben de desarrollar para cumplir con las necesidades de la empresa UBS.

ID	Enunciado de la Tarea	Nombre
BL001	Como *Gerente*, necesito poder limitar quienes pueden ingresar a la información de la empresa, con la finalidad de evitar que nuestra información vital tenga puntos de fuga	Manejo de usuarios
BL003	Como *vendedor*, necesito agregar nuevas cuentas de clientes, con la finalidad de llevar un mejor control de mis clientes.	Agregar cuentas clientes
BL004	Como *vendedor*, necesito ver todas las oportunidades de los clientes, con la finalidad de ver el avance y trabajo realizado día a día con los clientes	Listar oportunidades por cuentas
BL005	Como *vendedor*, necesito ver detalles principales de las cuentas clientes en cada oportunidad, con la finalidad de tener fácil acceso a su información de contacto	Detalles de cuenta por oportunidad
BL006	Como *vendedor*, deseo poder agregar información de los contactos de las cuentas, con la finalidad de tener más de una persona con la cual contactarme en la empresa	Crear contactos

ID	Enunciado de la Tarea	Nombre
	cliente	
BL007	Como *supervisor*, necesito buscar clientes según ciertos criterios, con la finalidad de ubicar clientes que se encuentren en ciertos estados de proceso	Filtrar clientes
BL009	Como un *vendedor*, necesito crear nuevas oportunidades para un cliente, con la finalidad de iniciar un nuevo proceso de venta con un cliente nuevo o existente	Crear oportunidades
BL010	Como *Gerente de informática*, necesito que el sistema cree un nuevo ambiente de trabajo por cada empresa que se registra para no estar creando copias del sistema original y así mantener el sistema desde un solo lugar.	multi empresas
BL011	Como *vendedor*, necesito poder manejar información específica de cada cuenta de clientes, con la finalidad de manejar la mayor cantidad de información que ayude a cerrar las ventas	Campos personalizados
BL012	Como *vendedor*, necesito poder llevar notas sobre los procesos que se realizan con cada cliente, con la finalidad de tener un registro de los detalles y avances del proceso de venta	Notas de clientes
BL013	Como un *vendedor*, necesito poder manejar documentos digitales de cada cuenta de cliente, con la finalidad de tener registro de los documentos generados en los procesos de Lead, oportunidad y cuenta	manejar archivos
BL014	Como un *vendedor*, necesito agregar y manejar tareas para cada cuenta, con el fin de tener un detalle del plan de acción seguido con cada cliente	Tareas en cuentas de clientes

ID	Enunciado de la Tarea	Nombre
BL015	Como *vendedor*, necesito poder ver un detalle de cada cuenta cliente, con el fin de obtener toda la información disponible del cliente como tareas, oportunidades, contactos, detalles generales.	Detalle de cuenta cliente
BL016	Como *administrador*, necesito poder cambiar el propietario de una cuenta de cliente, con la finalidad de poder distribuir la carga laboral entre los vendedores	Cambiar propietario de cuenta cliente
BL017	Como *vendedor*, necesito poder crear y manejar cuentas Leads, con la finalidad de llevar nuevos clientes a la empresa y tener un control de los mismos	Cuentas Leads
BL018	Como *vendedor*, necesito cambiar la etapa de una oportunidad por otra, con el fin de llevar un control actualizado de las fases de todas mis cuentas activas	Cambio de etapa de oportunidad
BL019	Como *vendedor*, necesito actualizar el tipo de cuenta de un cliente entre Leads, Oportunidad, Cuenta cerrada, con el fin de segmentar las cuentas según el proceso que se lleva con el cliente	Actualizar tipo de cuenta Lead/Oportunidad/Cuenta cerrada
BL021	Como *vendedor*, necesito determinar si una oportunidad ha sido ganada o perdida, con la finalidad de mostrar un resultado sobre mis gestiones y avanzar en los procesos	Oportunidad ganada/perdida
BL024	Como *supervisor*, necesito saber el calendario de citas de los vendedores y que estos puedan agregarlas al sistema, con el fin de conocer el estado del seguimiento a sus clientes	Calendario de citas
BL026	Como *vendedor*, necesito ver el listado de contactos de un cliente en los detalles de cuenta, con la finalidad de ingresar sin problemas a los detalles de cada uno	Listar contactos en cuentas cliente

ID	Enunciado de la Tarea	Nombre
BL027	Como *vendedor*, necesito poder ver el listado de todas las tareas para con el cliente dentro del detalle de su cuenta, con la finalidad de saber que pendientes existen con el mismo	Listado de tarea en cuentas cliente
BL029	Como *administrador*, me gustaría poder manejar quien puede o no tener acceso a información específica del sistema, con el objetivo de poder dividir las tareas del equipo.	Roles de usuarios
BL031	Como *supervisor*, necesito poder ver un reporte de cuentas, oportunidades, leads y otros con la finalidad de brindar detalles de nuestras operaciones a nuestros superiores	Reportes
BL032	Como *supervisor*, debo de poder distribuir a los vendedores por territorios o sectores, con la finalidad de hacer una mejor distribución y mayor alcance de clientes.	Territorios
BL033	Como *administrador*, necesito asignar a cada empleado un departamento y área laboral, con el fin de mantener a cada uno relacionado a su área correspondiente	Departamentos y áreas laborales
BL034	Como *administrador*, necesito poder crear un listado de todos nuestros empleados enlazada a sus operaciones dentro del sistema, con la finalidad de mantener una lista actualizada de los mismos	Listado de empleados

Tabla 2 Product Backlog

5.2.2. Planificación de los Sprint

Durante la reunión de planificación se han organizado cada una de las tareas del Backlog inicial otorgando a cada una prioridad de entrega, de esta manera hemos obtenido un total de cuatro Sprints debidamente equilibrados como se muestra a continuación.

SPRINT #1			
Semana de inicio #	**3**	Semana de fin #	**5**
Revisiones de avances			
Scrum Manager	**Diario, 8:30 am**	Product Owner	**Semanal, Lunes 8:00am**
Tareas a desarrollar			
PB ID	**Prioridad**	**Nombre**	
BL003	Alta	Agregar cuentas clientes	
BL009	Alta	Crear oportunidades	
BL001	Alta	Manejo de usuarios	
BL006	Media	Crear contactos	
BL007	Media	Filtrar clientes	
BL004	Baja	Listar oportunidades por cuentas	
BL005	Baja	Detalles de cuenta por oportunidad	

Tabla 3 Definición Sprint #1

SPRINT #2			
Semana de inicio #	**6**	**Semana de fin #**	**9**
Revisiones de avances			
Scrum Manager	**Diario, 8:30 am**	Product Owner	**Semanal, Lunes 8:00am**
Tareas a desarrollar			
PB ID	**Prioridad**	**Nombre**	
BL010	Alta	Multi empresas	
BL013	Media	Manejar archivos	
BL014	Media	Tareas en cuentas de clientes	
BL015	Media	Detalle de cuenta cliente	
BL012	Baja	Notas de clientes	

SPRINT #2			
Semana de inicio #	**6**	**Semana de fin #**	**9**
Revisiones de avances			
Scrum Manager	**Diario, 8:30 am**	Product Owner	**Semanal, Lunes 8:00am**
Tareas a desarrollar			
BL016	baja	Cambiar propietario de cuenta cliente	
BL018	baja	Cambio de etapa de oportunidad	

Tabla 4 Definición Sprint #2

SPRINT #3			
Semana de inicio #	**10**	**Semana de fin #**	**14**
Revisiones de avances			
Scrum Manager	**Diario, 8:30 am**	Product Owner	**Semanal, lunes 8:00am**
Tareas a desarrollar			
PB ID	**Prioridad**	**Nombre**	
BL017	Alta	Cuentas Leads	
BL019	Media	Actualizar tipo de cuenta Lead/Oportunidad/Cuenta cerrada	
BL024	Media	Calendario de citas	
BL021	Baja	Oportunidad ganada/perdida	
BL026	Baja	Listar contactos en cuentas cliente	
BL027	Baja	Listado de tarea en cuentas cliente	
BL032	Baja	Territorios	

Tabla 5 Definición Sprint #3

SPRINT #4			
Semana de inicio #	**15**	**Semana de fin #**	**18**
Revisiones de avances			
Scrum Manager	**Diario, 8:30 am**	Product Owner	**Semanal, lunes 8:00am**
Tareas a desarrollar			
PB ID	**Prioridad**	**Nombre**	
BL031	Alta	Reportes	
BL011	Media	Campos personalizados	

SPRINT #4			
Semana de inicio #	**15**	**Semana de fin #**	**18**
Revisiones de avances			
Scrum Manager	**Diario, 8:30 am**	Product Owner	**Semanal, lunes 8:00am**
Tareas a desarrollar			
BL022	Media	Panel de administración	
BL029	Media	Roles de usuarios	
BL033	Baja	Departamentos y áreas laborales	
BL034	Baja	Listado de empleados	

Tabla 6 Definición Sprint #4

5.2.3. Diseño del sistema

CRMX utiliza la arquitectura de software como un servicio (SaaS por sus siglas en inglés) donde varias empresas (tenants) comparten la misma instancia para almacenar sus datos como se muestra en la Ilustración 2. Esta instancia suele estar dividida (o particionada) para evitar que las empresas accedan a la información de las demás. Esto es como un edificio de gran altura donde los planos de planta generalmente se establecen, pero se pueden hacer pequeños cambios estéticos en las unidades individuales.

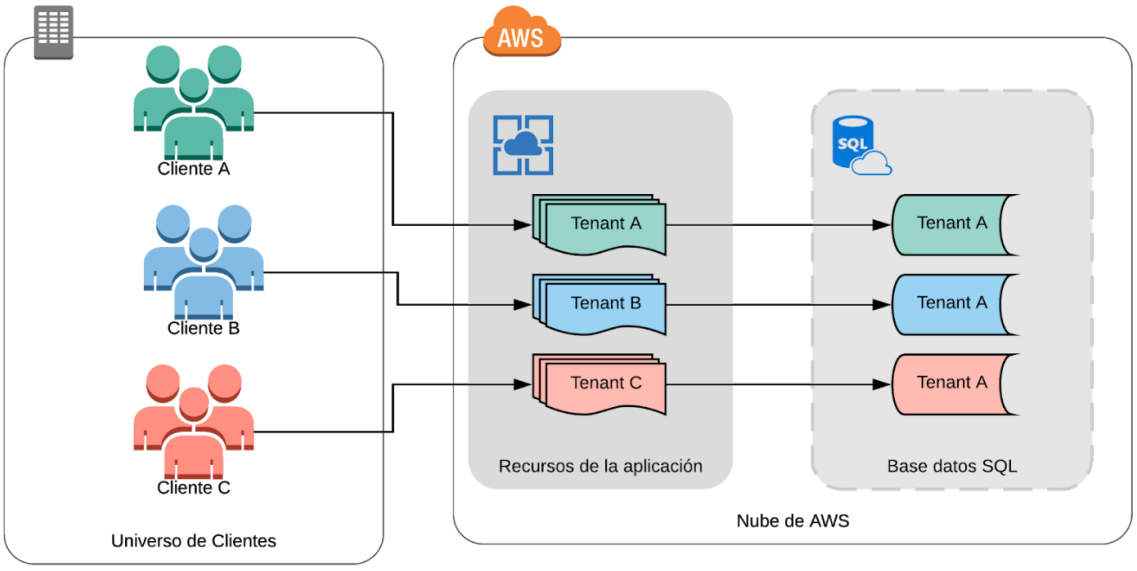

Ilustración 2 Estructura de la aplicación

La base de datos utiliza un enfoque de diseño de tablas compartidas en el mismo esquema. La estructura del modelo de los tenant posee datos en común (datos de referencia, metadatos, Masterdata) pero que también son específicos para cada usuario del tenant (por ejemplo, datos maestros, valores de búsqueda / gestión de listas de selección, nuevos campos / columnas de datos configurados / ampliados). El modelo de implementación utiliza el enfoque de instancia única compartida y base de datos compartida como se muestra en la Ilustración 3.

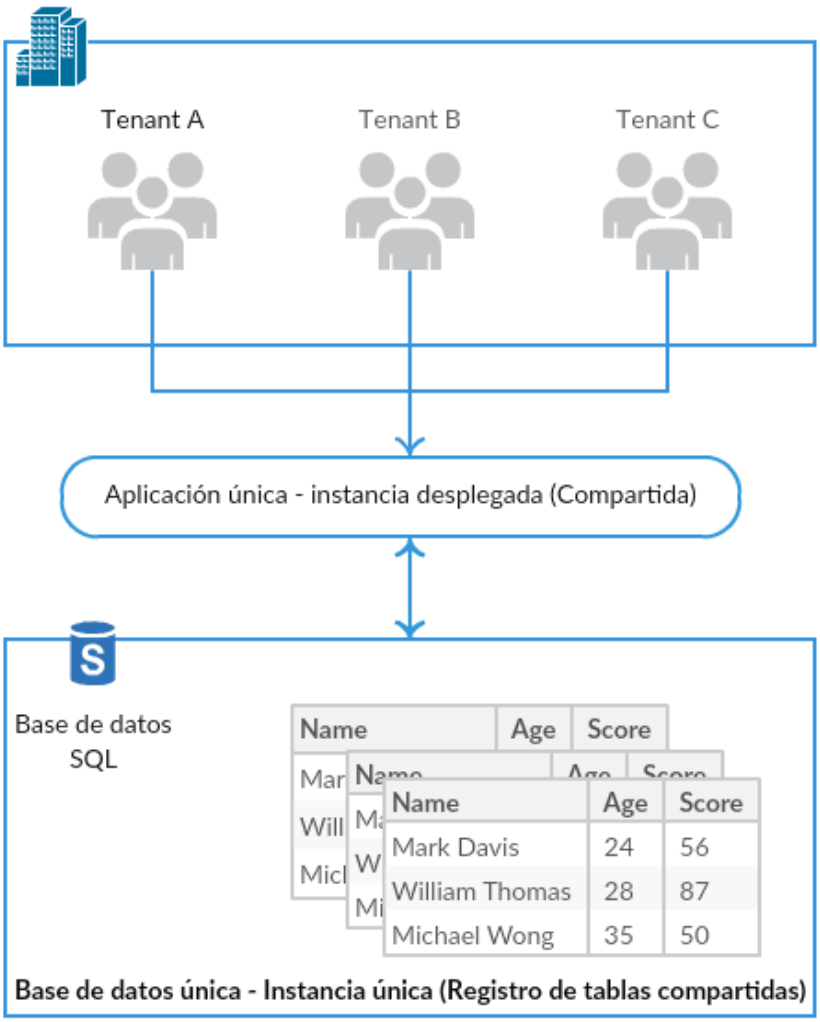

Ilustración 3 Modelo de instancia única compartida

La aplicación consta de varios componentes, incluida la aplicación web CRMX, que implementa la interfaz de usuario, junto con algunos servicios de back-end como File System Service (FSS) encargado del manejo de los archivos (PDF, Excel, Word e imágenes.) que servirán de soporte de documentación generada

para cada uno de los clientes, los servicios API de CRMX que constituyen el núcleo de la lógica empresarial de la aplicación y el servicio de administración de gestión de seguridad (SMA) que es el responsable de la seguridad.

Como se representa en la Ilustración 4 CRMX utiliza una arquitectura que estructura la aplicación como un conjunto de servicios de colaboración debidamente acoplados. Cada servicio implementa un conjunto de funciones estrechamente relacionadas. Los Servicios se comunican utilizando protocolos síncronos como HTTP / REST. Los servicios pueden ser desarrollados e implementados independientemente unos de otros. Cada servicio tiene su propia base de datos para ser desconectado de otros servicios.

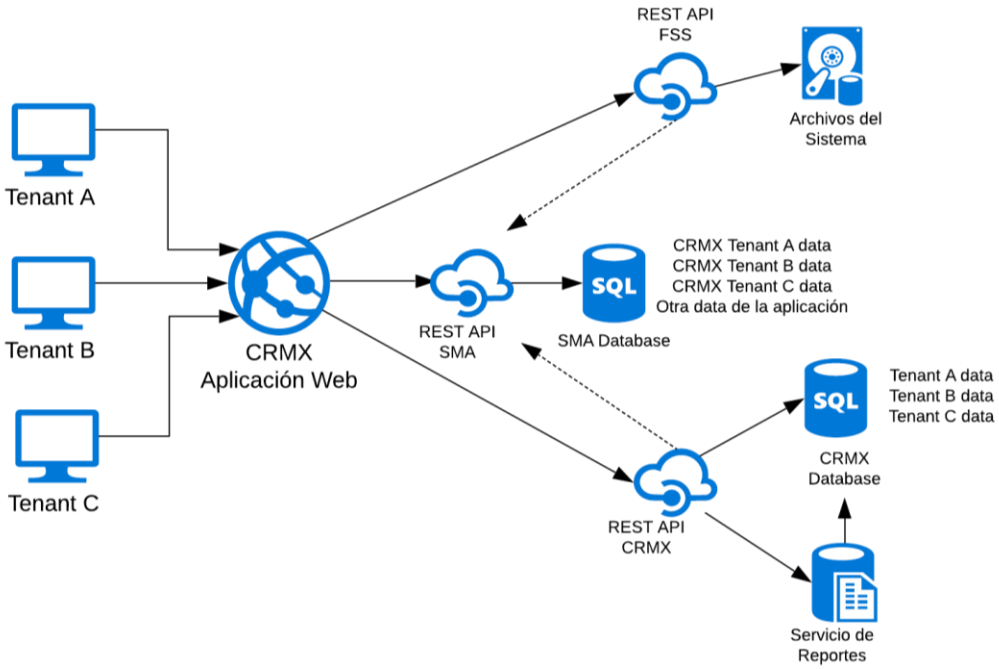

Ilustración 4 Arquitectura de la aplicación

CRMX usa una estrategia de etiquetado en blanco parcial para detectar al tenant, como resultado podríamos distinguir el subdominio de la solicitud (por ejemplo, testcompany.crmxcloud.com). Para garantizar esto en las configuraciones de nuestro dominio utilizaremos un registro de subdominio comodín (*) dentro de los ajustes del dominio crmxcloud.com para obtener el enfoque anterior.

5.2.3.1. Diagrama de componentes

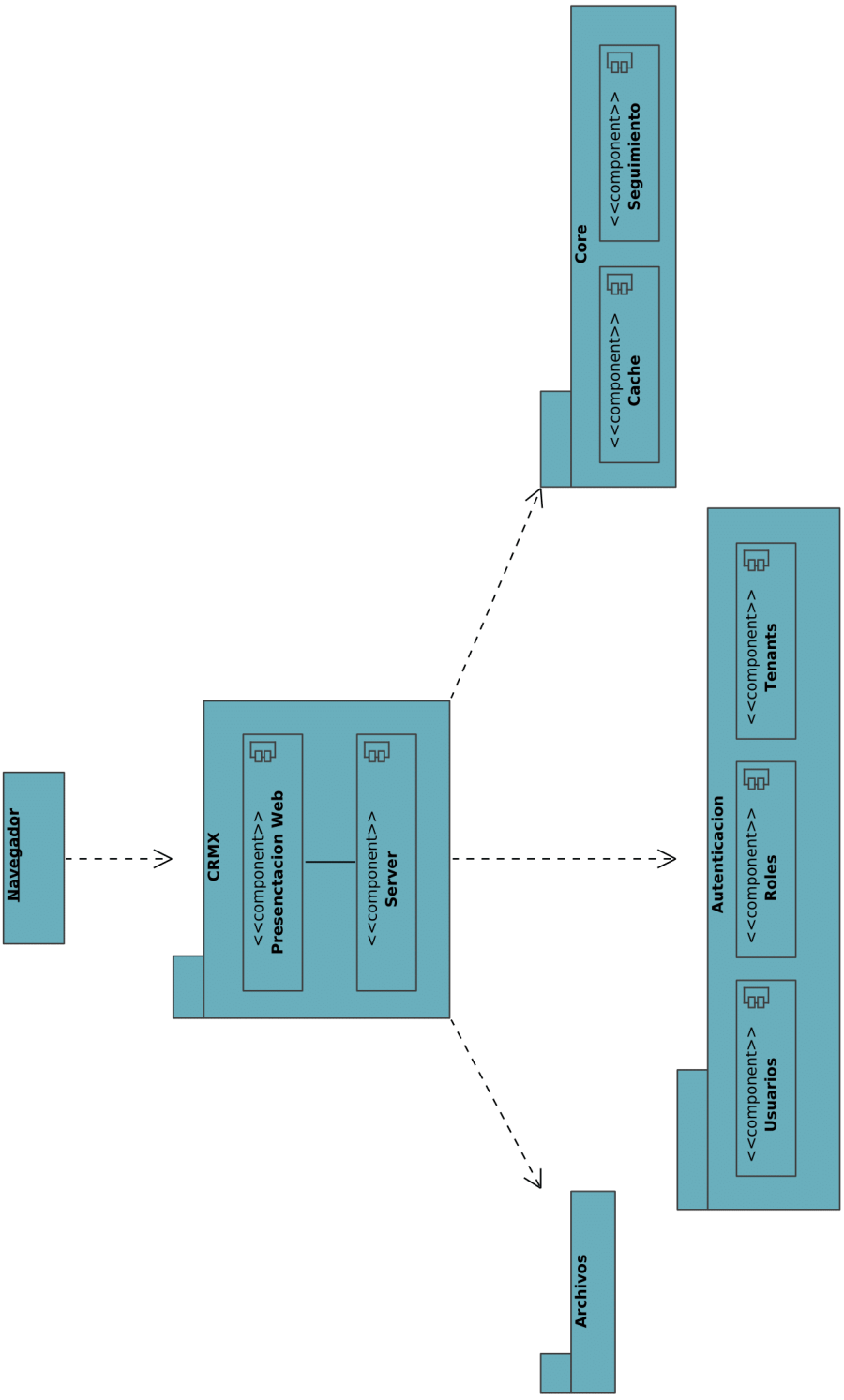

Ilustración 5 Diagrama de componentes

5.2.3.2. Mapa de navegación del sitio

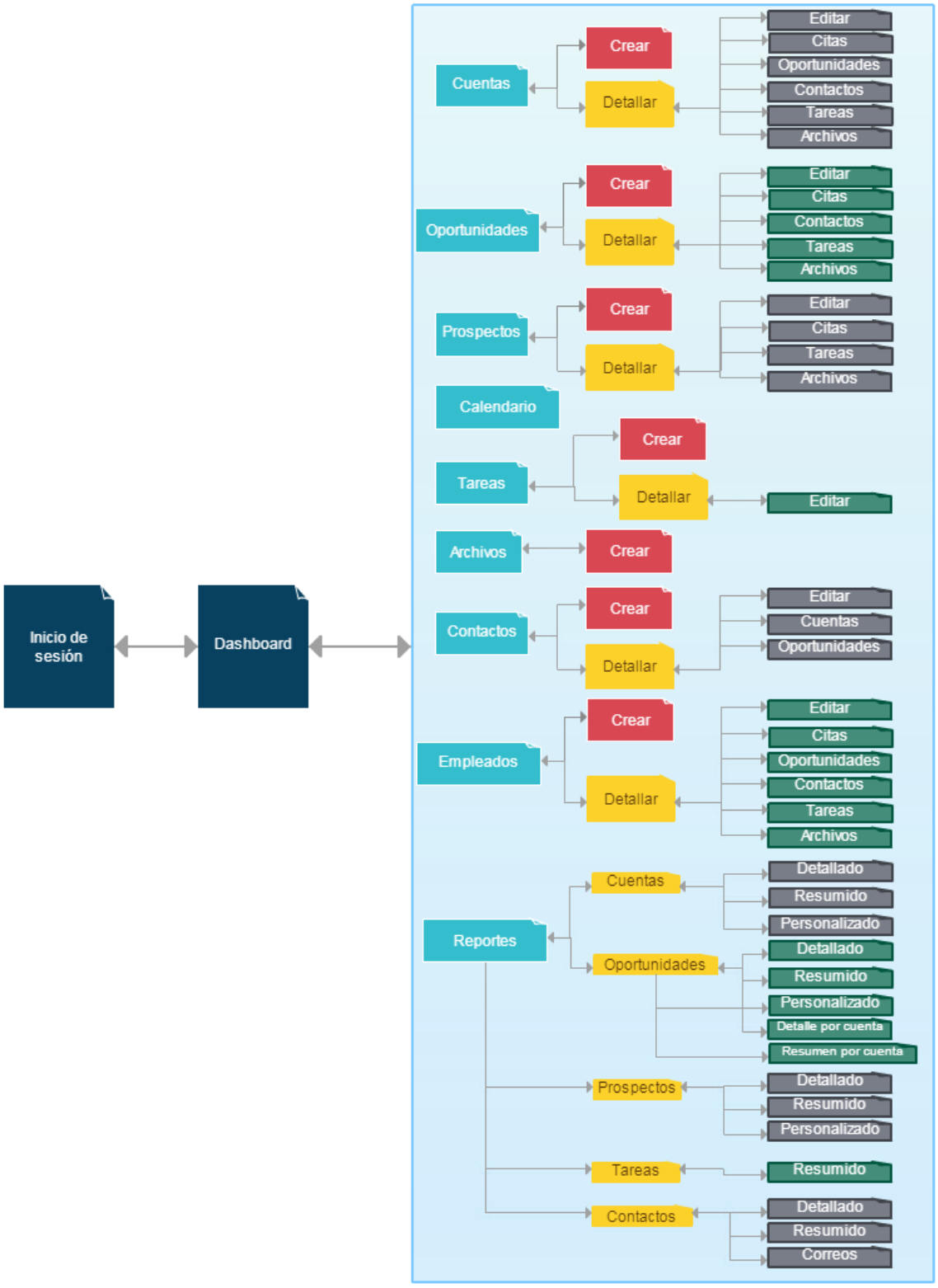

Ilustración 6 Mapa principal del sitio

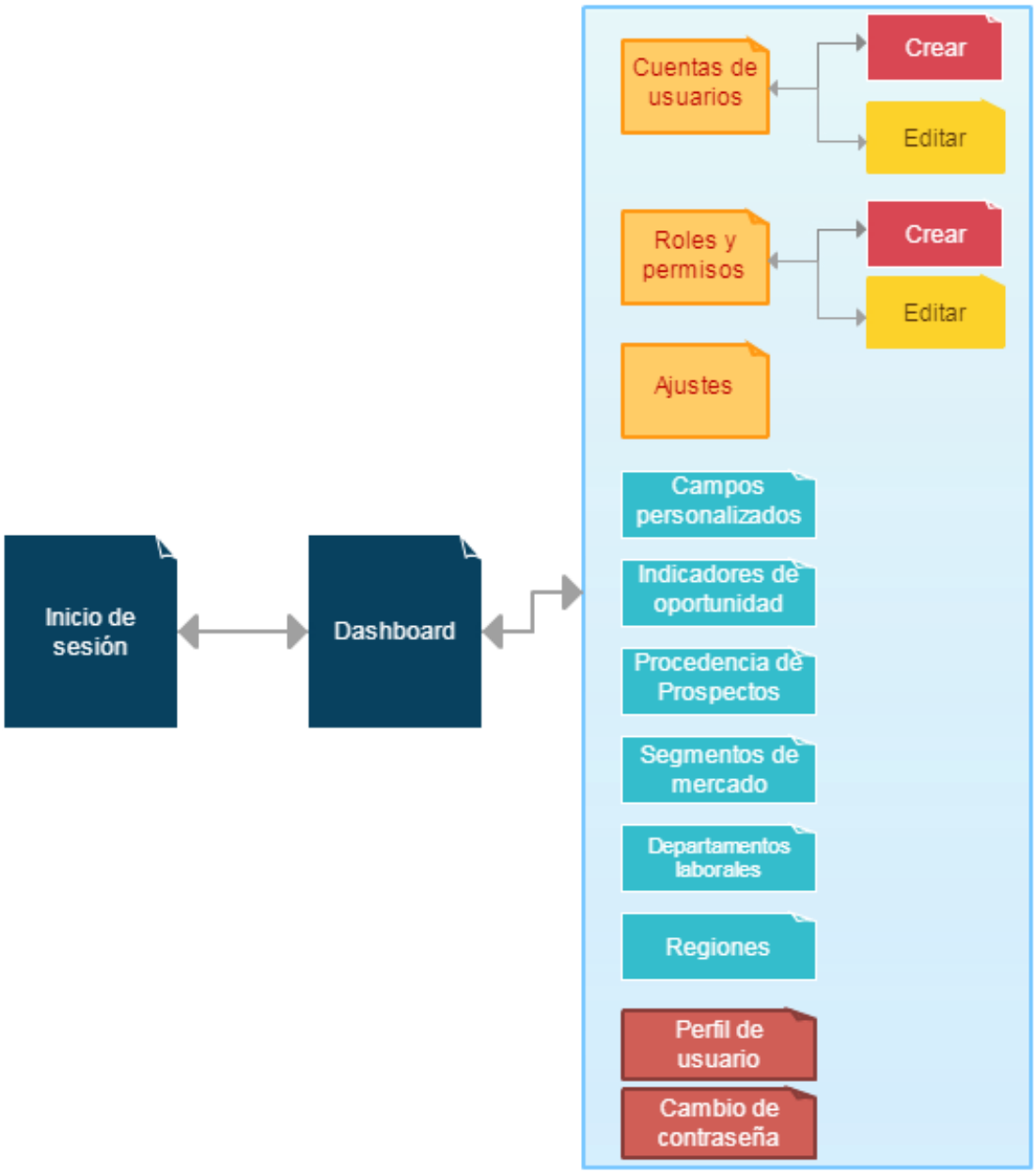

Ilustración 7 Mapa del sitio - Ajustes

5.2.3.3. Modelo Conceptual de la Base de Datos CRM y SMA

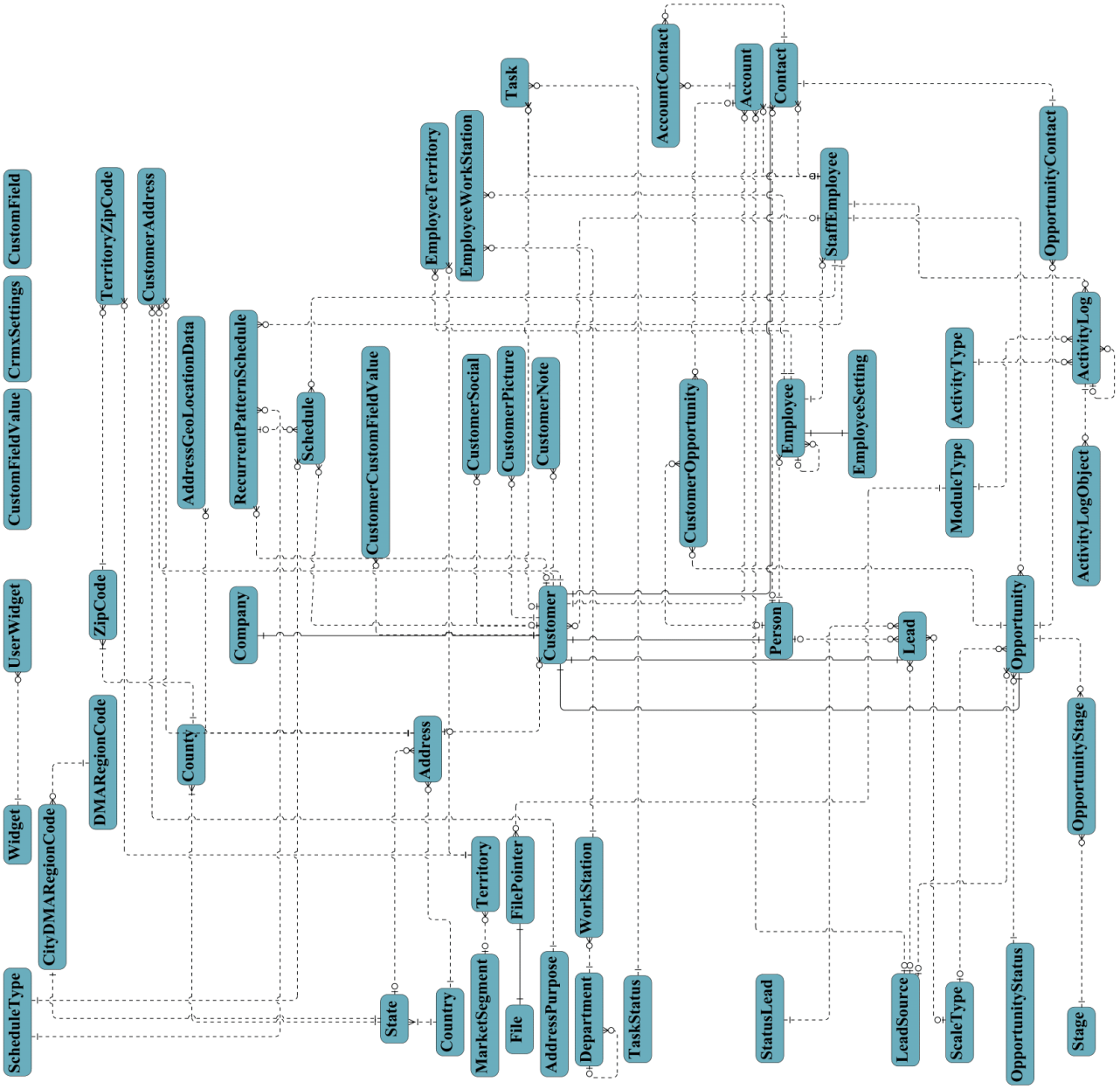

Ilustración 8 Modelo Conceptual Base de Datos CRM

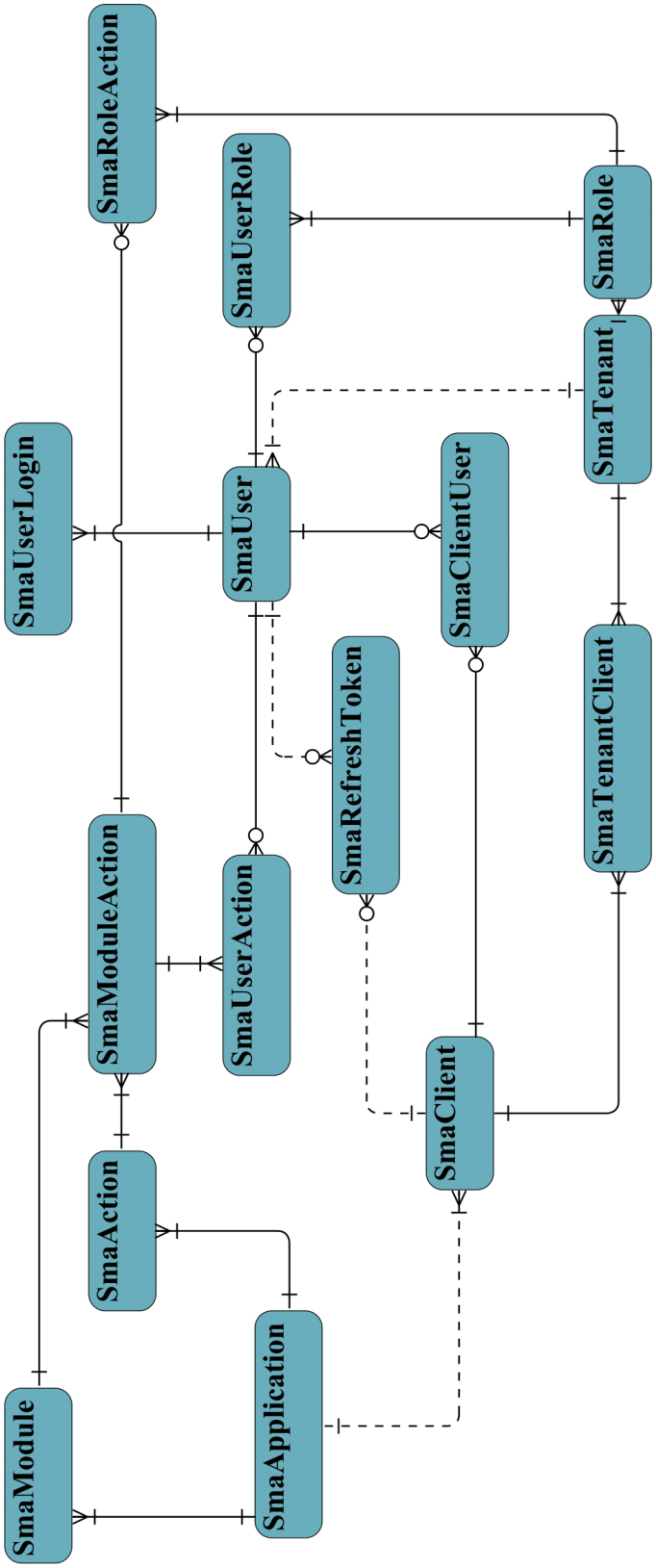

Ilustración 9 Modelo Conceptual de la Base de Datos SMA

5.2.3.4. Diseño de estructura del sitio web

Los siguientes Wireframe representan el bosquejo de la estructura y distribución del sitio web, tomando en cuenta los elementos visuales característicos de los sistemas anteriormente en uso, mejorando su contenido, presentación y organización con la opinión del usuario final.

En la Ilustración 10 se muestra la representación esquematica de la pantalla principal (Dashboard) del sitio web donde a elección y administración del usuario se listará un grupo de widgets que cuentan con un resumen de datos sobre sus avances representado en estadisticas, resumen de tareas y otros.

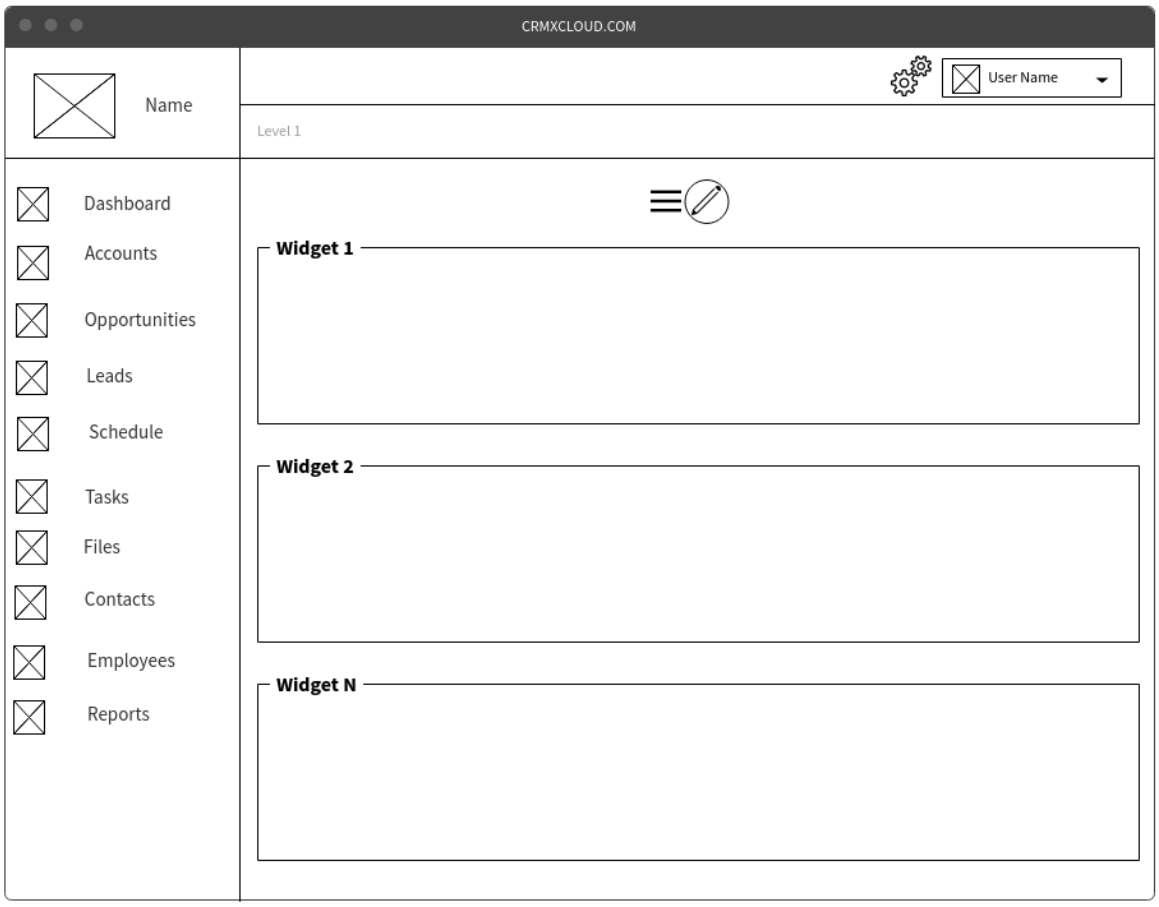

Ilustración 10 Wireframe - Dashboard

La Ilustración 11 nos permite observar cómo se espera que sean mostrados los resultados obtenidos al momento de entrar a las categorías Accounts, Opportunities, Leads, Tasks, Files, Contacts y Employees.

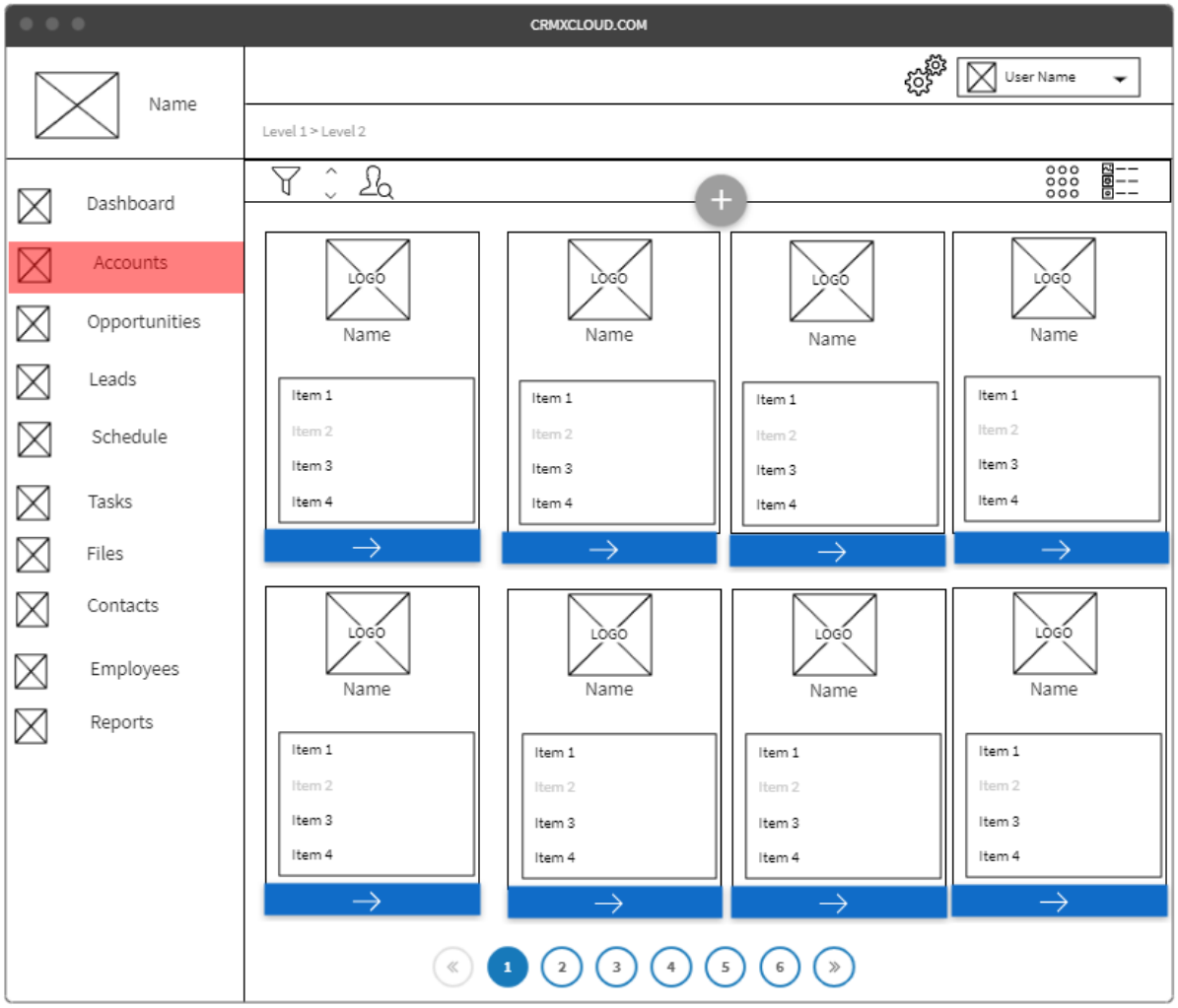

Ilustración 11 Wireframe - listar resultados

Ilustración 12 Wireframe - Agregar/Editar registros

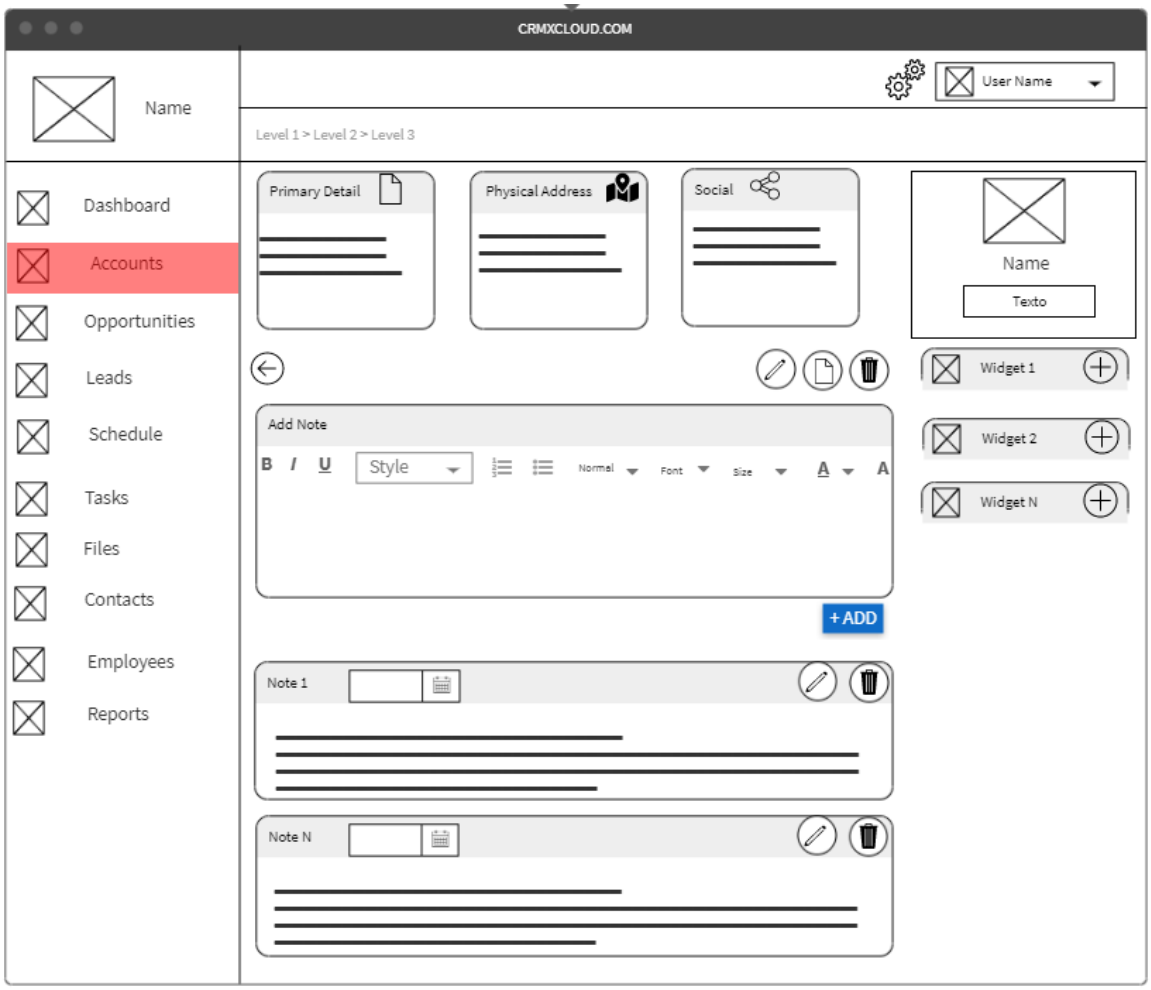

Ilustración 13 Wireframe – Detalle de Cuentas, Oportunidades y Leads

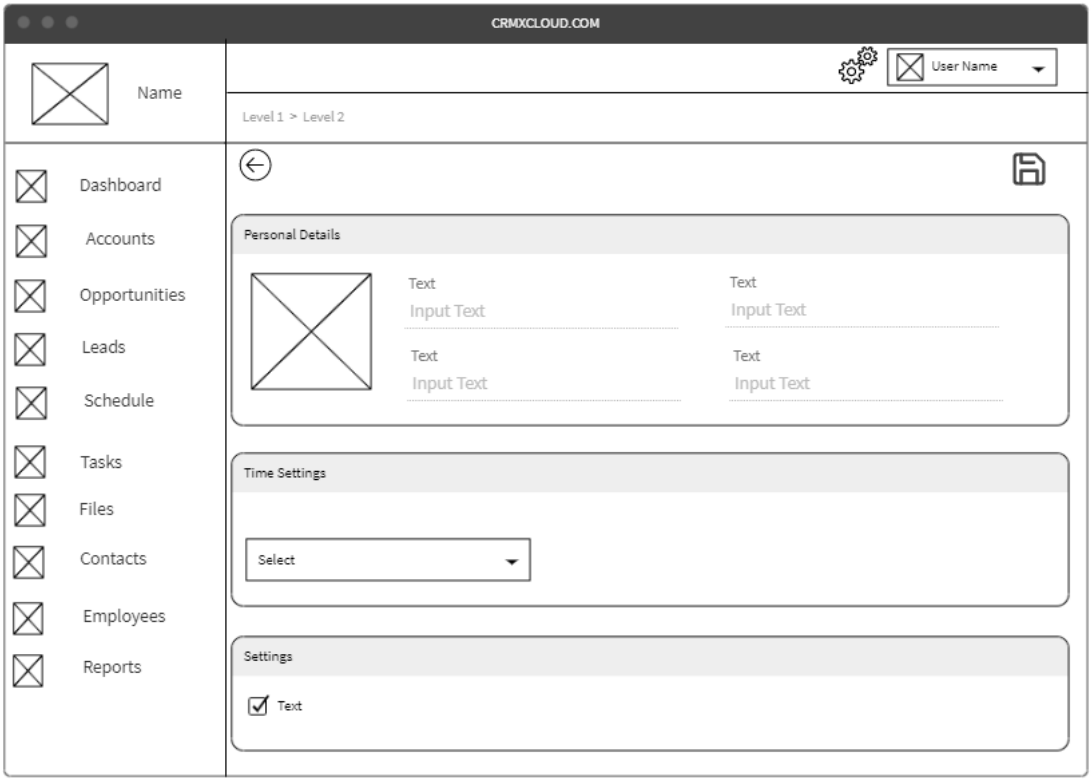

Ilustración 14 Wireframe - Ajustes de perfil de usuario

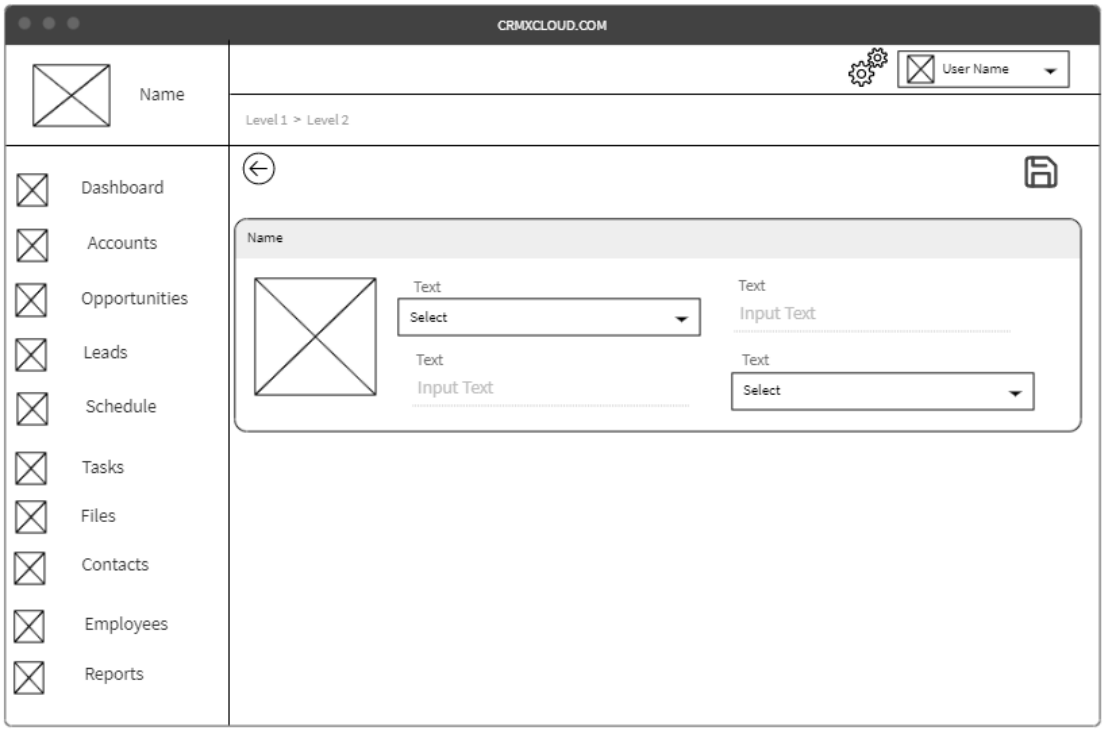

Ilustración 15 Wireframe - Ajustes de perfil tenant

5.2.4. Desarrollo del Sprint #1

Basados en el análisis y diseño del sistema se procede al desarrollo del mismo siguiendo los compromisos adquiridos para cada Sprint; a continuación, demostraremos los resultados obtenidos por cada uno de ellos.

5.2.4.1. Tareas y sub-tareas desarrolladas

SPRINT #1					
Semana de inicio #		**3**		**Semana de fin #**	**5**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
BL003	Agregar cuentas clientes	Hecho	BLS001	Diseñar y desarrollar pantallas de cuentas	Hecho
			BLS002	Crear y enlazar tablas de base de datos de cuentas	Hecho
			BLS003	Preparar y ejecutar caso de prueba	Hecho
BL009	Crear oportunidades	Hecho	BLS004	Diseñar y desarrollar pantallas de oportunidades	Hecho
			BLS005	Crear y enlazar tablas de base de datos de oportunidades	Hecho
			BLS006	Preparar y ejecutar caso de prueba	Hecho
BL001	Manejo de usuarios	Hecho	BLS007	Diseñar y desarrollar pantalla de login	Hecho
			BLS008	Crear y enlazar tabla de usuarios	Hecho

SPRINT #1					
Semana de inicio #		**3**		**Semana de fin #**	**5**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
			BLS009	Crear algoritmos de validación y tokens de seguridad	Hecho
			BLS010	Preparar y ejecutar caso de prueba	Hecho
BL006	Crear contactos	Hecho	BLS011	Diseñar y desarrollar pantallas de Contactos	Hecho
			BLS012	Crear y enlazar tablas de base de datos de contactos	Hecho
			BLS013	Almacenar, actualizar, eliminar información de contactos y enlazarlos a cuentas	Hecho
BL007	Filtrar clientes	Hecho	BLS014	Agregar diseño de filtros a pantalla de cuentas	Hecho
			BLS015	Crear algoritmos de búsqueda por parámetros en cuentas	Hecho
			BLS016	Mostrar resultados de búsqueda en pantalla de cuentas	Hecho
BL004	Listar oportunidades	Hecho	BLS021	Crear pantalla detalles de cuentas	Hecho

SPRINT #1					
Semana de inicio #		**3**		**Semana de fin #**	**5**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
	por cuentas		BLS022	Crear campo en tabla oportunidades de relación a cuentas	Hecho
			BLS023	En detalle de cuenta mostrar enlaces a oportunidades enlazadas	Hecho
BL005	Detalles de cuenta por oportunidad	Hecho	BLS024	Diseñar y desarrollar pantalla de detalles de oportunidad	Hecho
			BLS025	En detalle de oportunidad mostrar información de contacto de la cuenta enlazada	Hecho

Tabla 7 Resultado de tareas y sub-tareas del Sprint #1

5.2.4.2. Base de datos SMA del Sprint

Tablas de datos creadas durante el desarrollo del sprint #1 para la base de datos SMA encargada de la gestión de seguridad del sistema. El diagrama completo de la base de datos se puede observar en la Ilustración 58 Modelo Físico de la Base de Datos SMA dentro de Anexos.

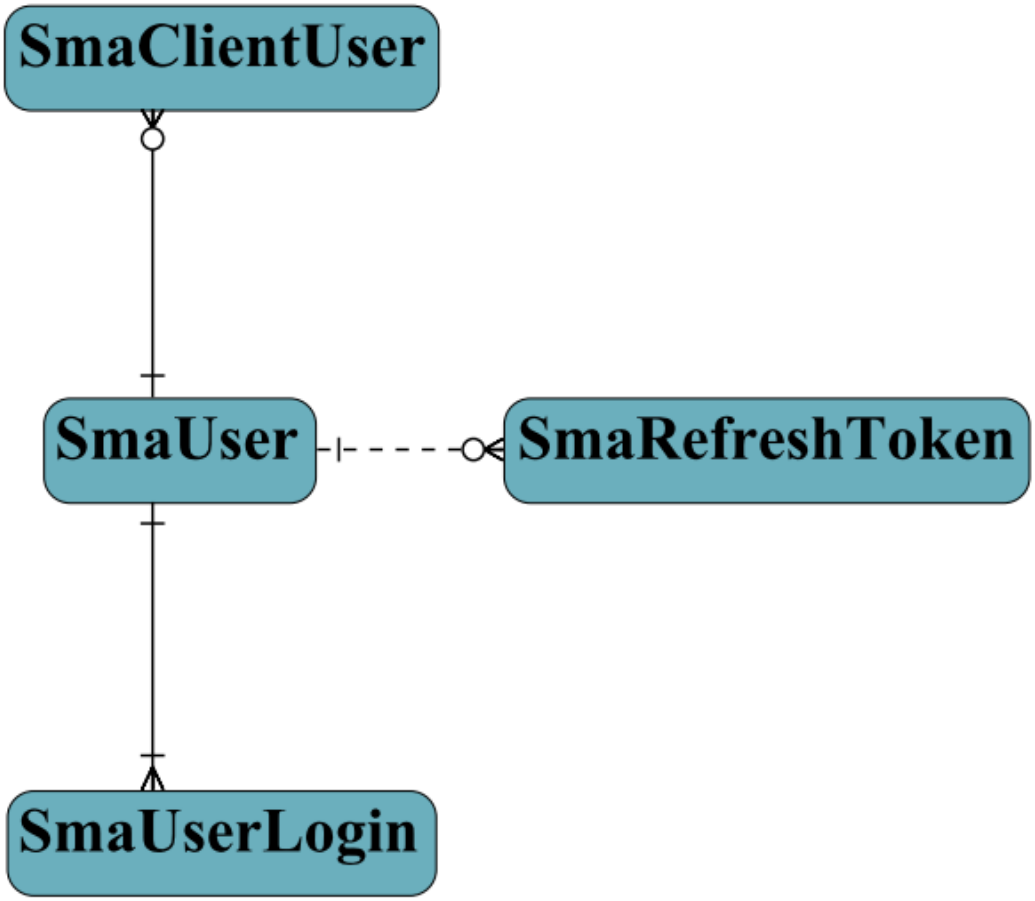

Ilustración 16 Tablas de datos SMA del Sprint #1

5.2.4.3. Base de datos CRM del Sprint

Tablas de datos creadas durante el desarrollo del sprint #1 para la base de datos CRM encargada de la gestión operacional del sistema. El diagrama completo de la base de datos se puede observar en la Ilustración 57 Modelo Físico Base de Datos CRM.

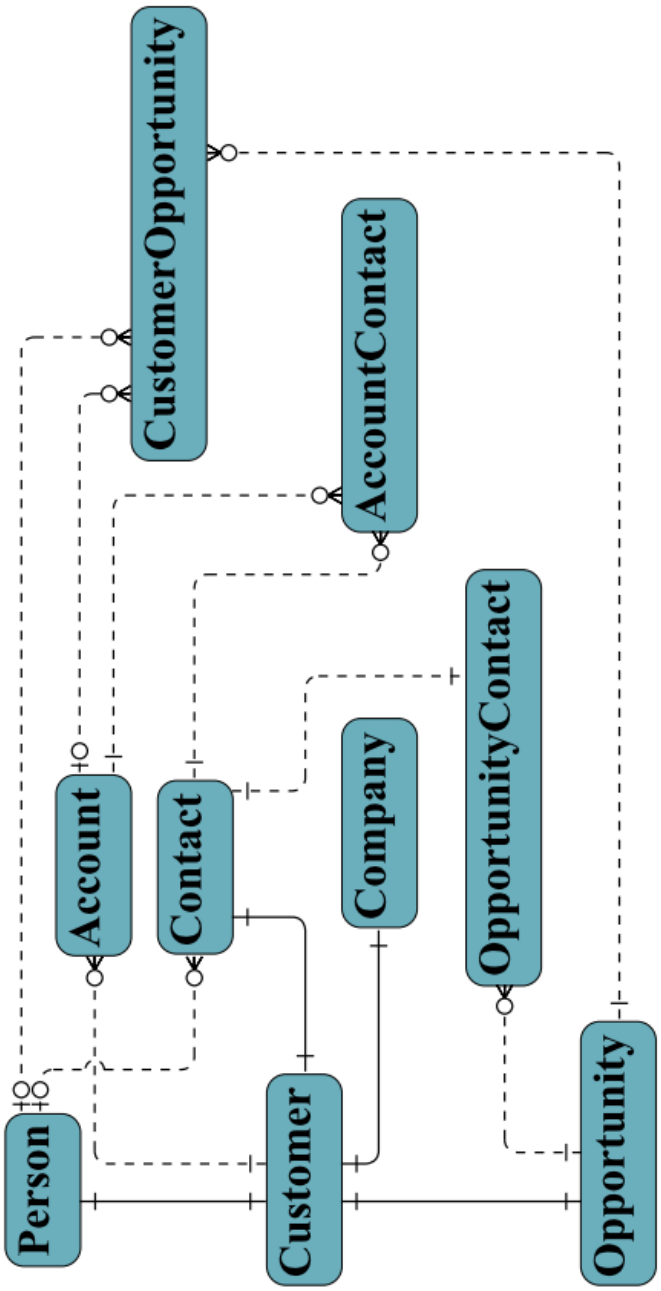

Ilustración 17 Tablas de datos del CRM del Sprint #1

5.2.4.4. Casos de Prueba del Sprint

PB ID: BL001

Nombre: Manejo de usuarios

Descripción: Crear un nuevo usuario, probar opción editar y por ultimo proceder a eliminar el usuario recién creado.

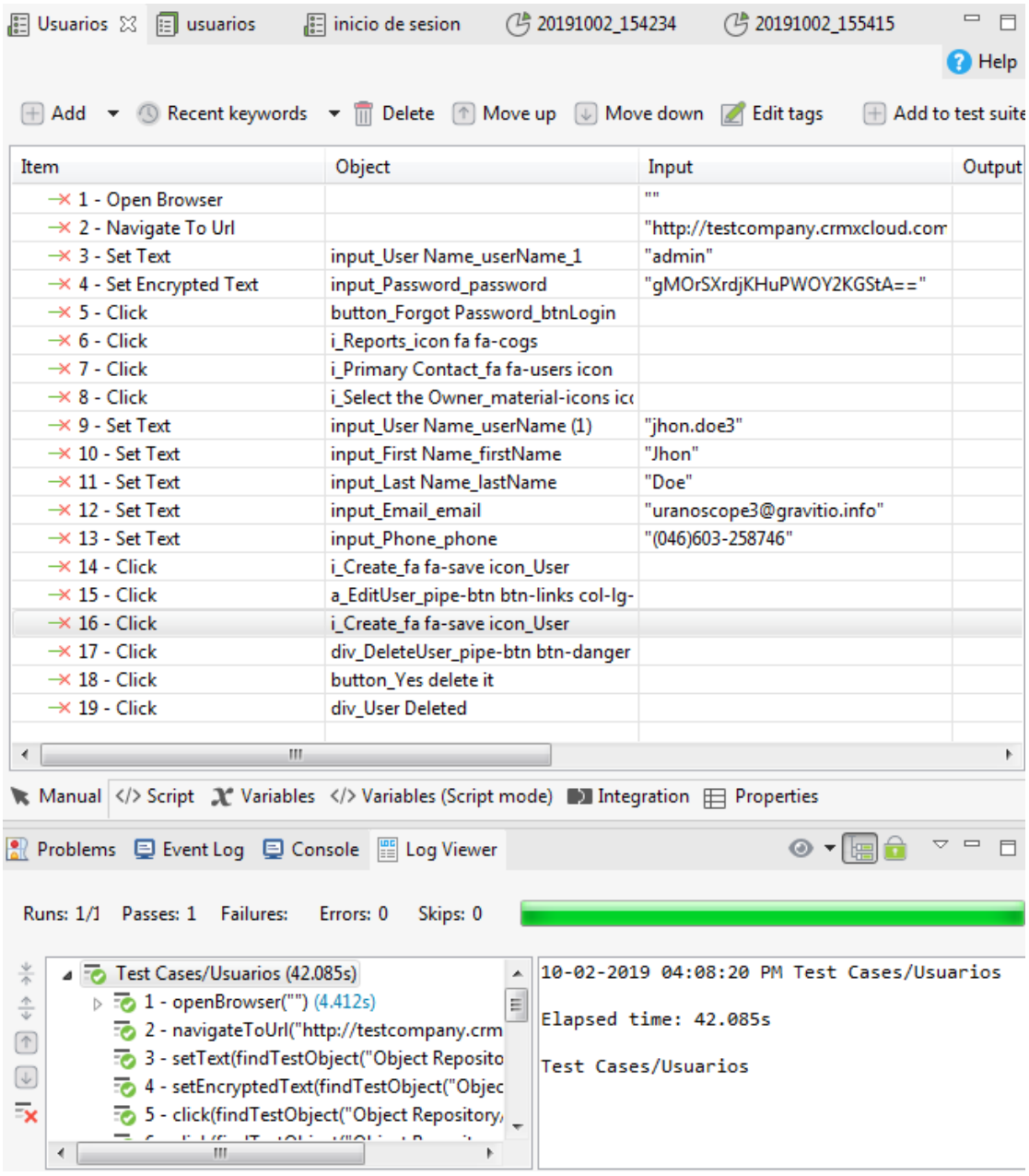

Ilustración 18 Caso de Prueba BL001 - Ingreso a la Aplicación

PB ID: BL003

Nombre: Agregar cuentas clientes

Descripción: Dirigirse a la sección de "*Cuentas*" luego tratar de crear una nueva cuenta cliente.

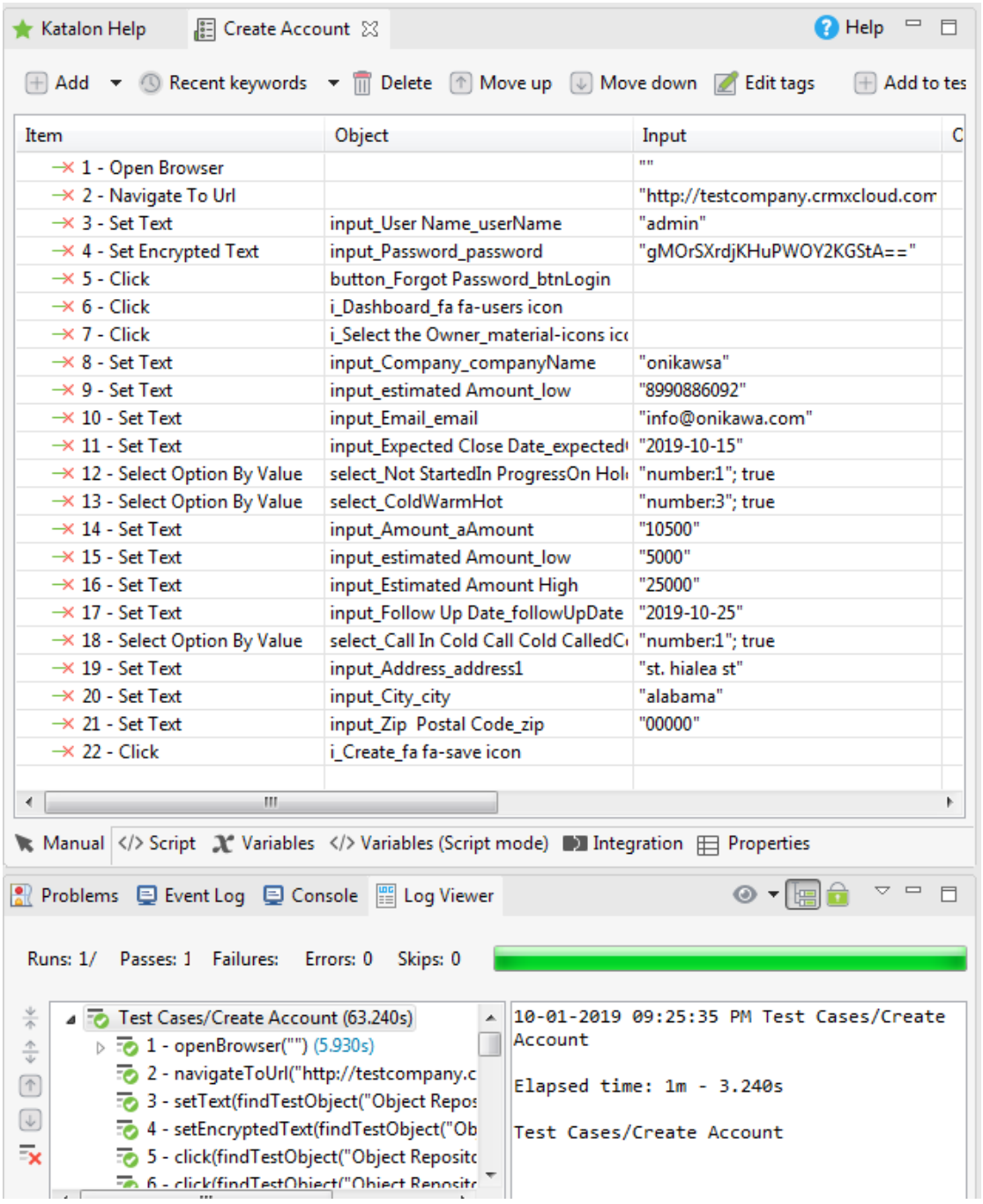

Ilustración 19 Caso de Prueba BL003 - Agregar Cuentas Clientes

PB ID: BL006

Nombre: Agregar contactos

Descripción: Ingresar a la aplicación, dirigirse a sección contactos, agregar un nuevo contacto, una vez agregado editarlo y por último se procede a eliminarlo.

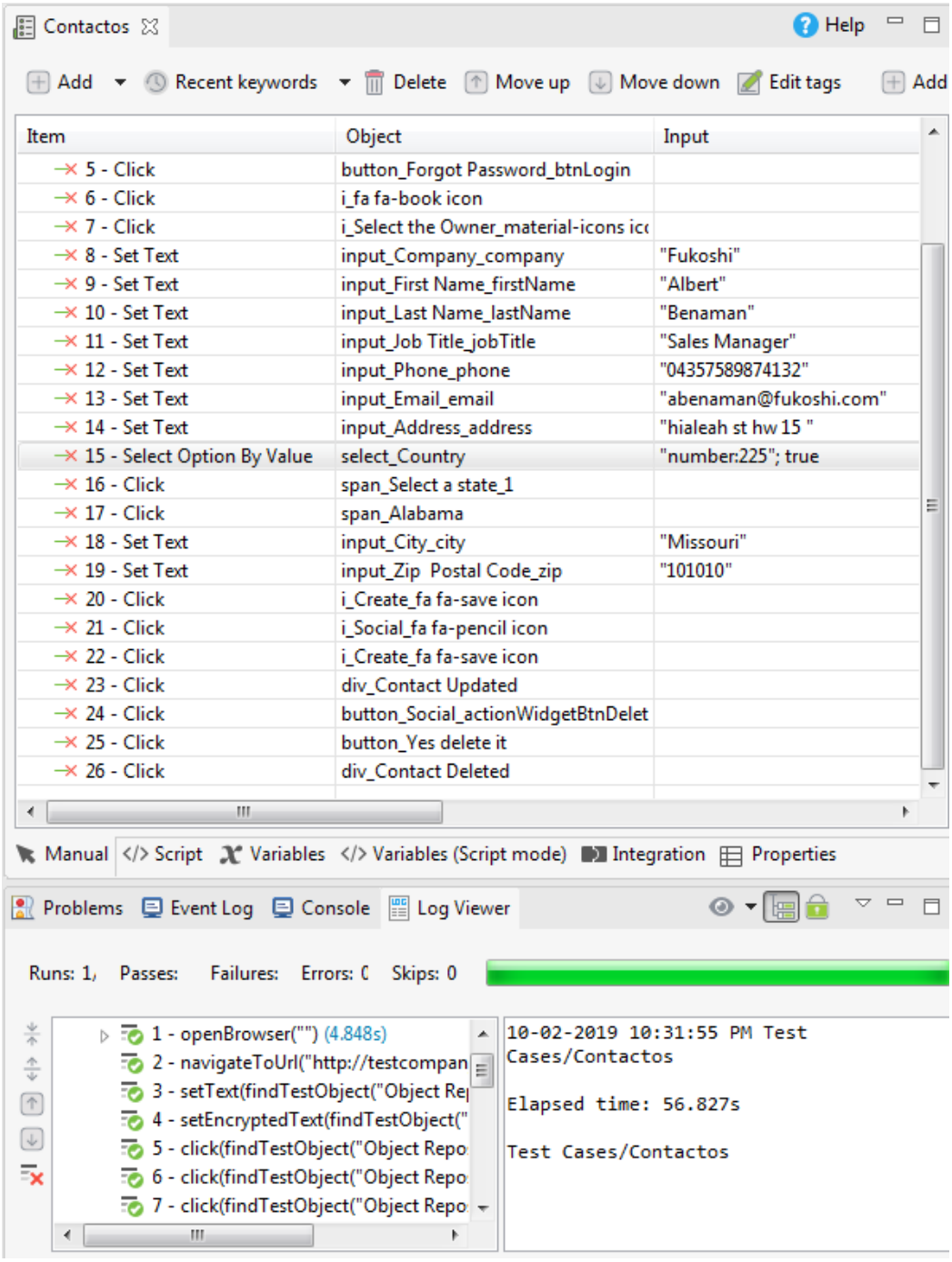

Ilustración 20 Caso de prueba BL006 - Crear Contactos

PB ID: BL009

Nombre: Agregar Oportunidad.

Descripción: Ingresar a sección oportunidades y agregar una nueva oportunidad de tipo Residencial.

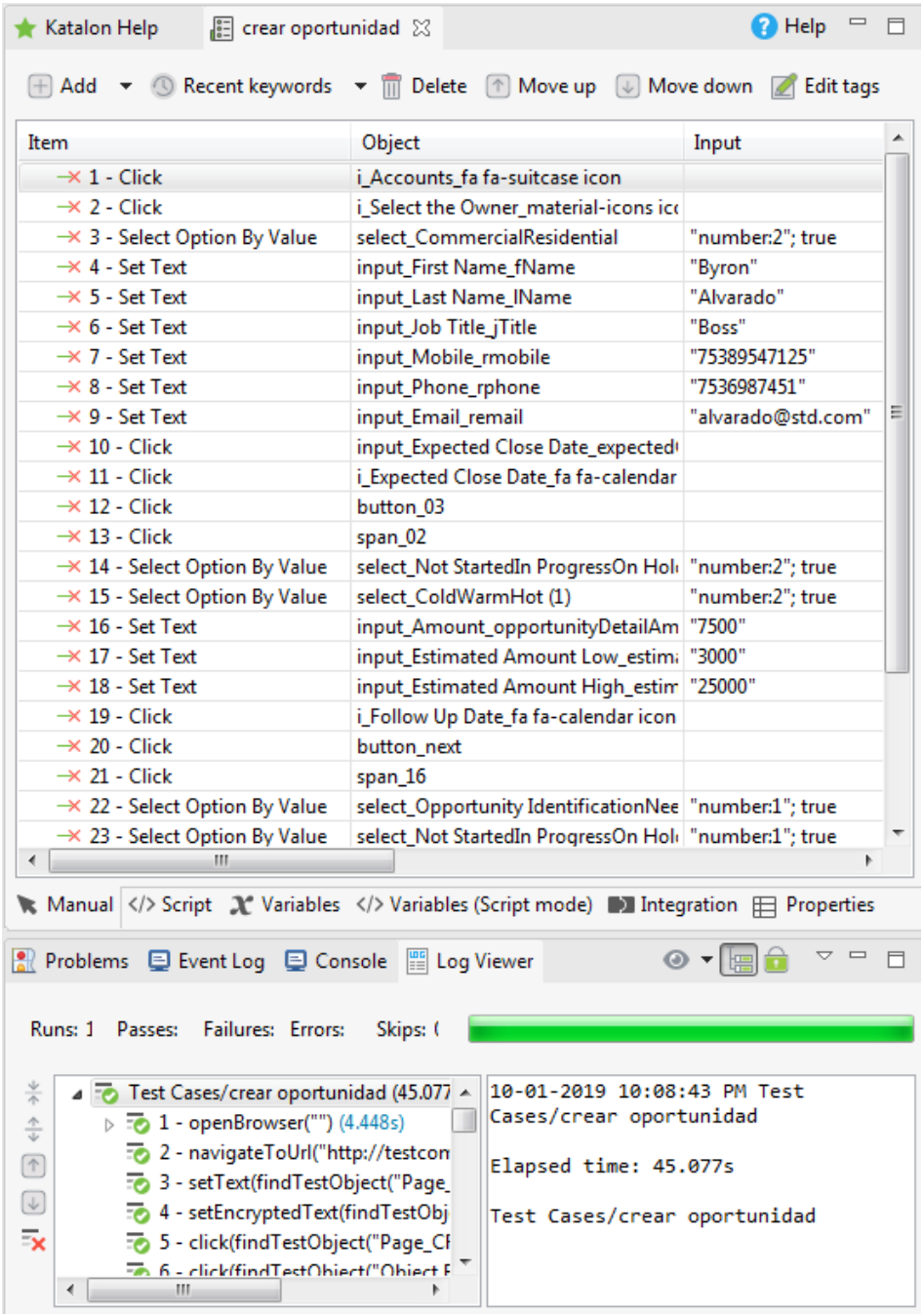

Ilustración 21 Caso de prueba BL009 - Agregar Oportunidades

PB ID: BL007

Nombre: Filtrar Clientes

Descripción: Dirigirse a la vista de todos los clientes luego utilizar los filtros para buscar clientes de tipo comercial.

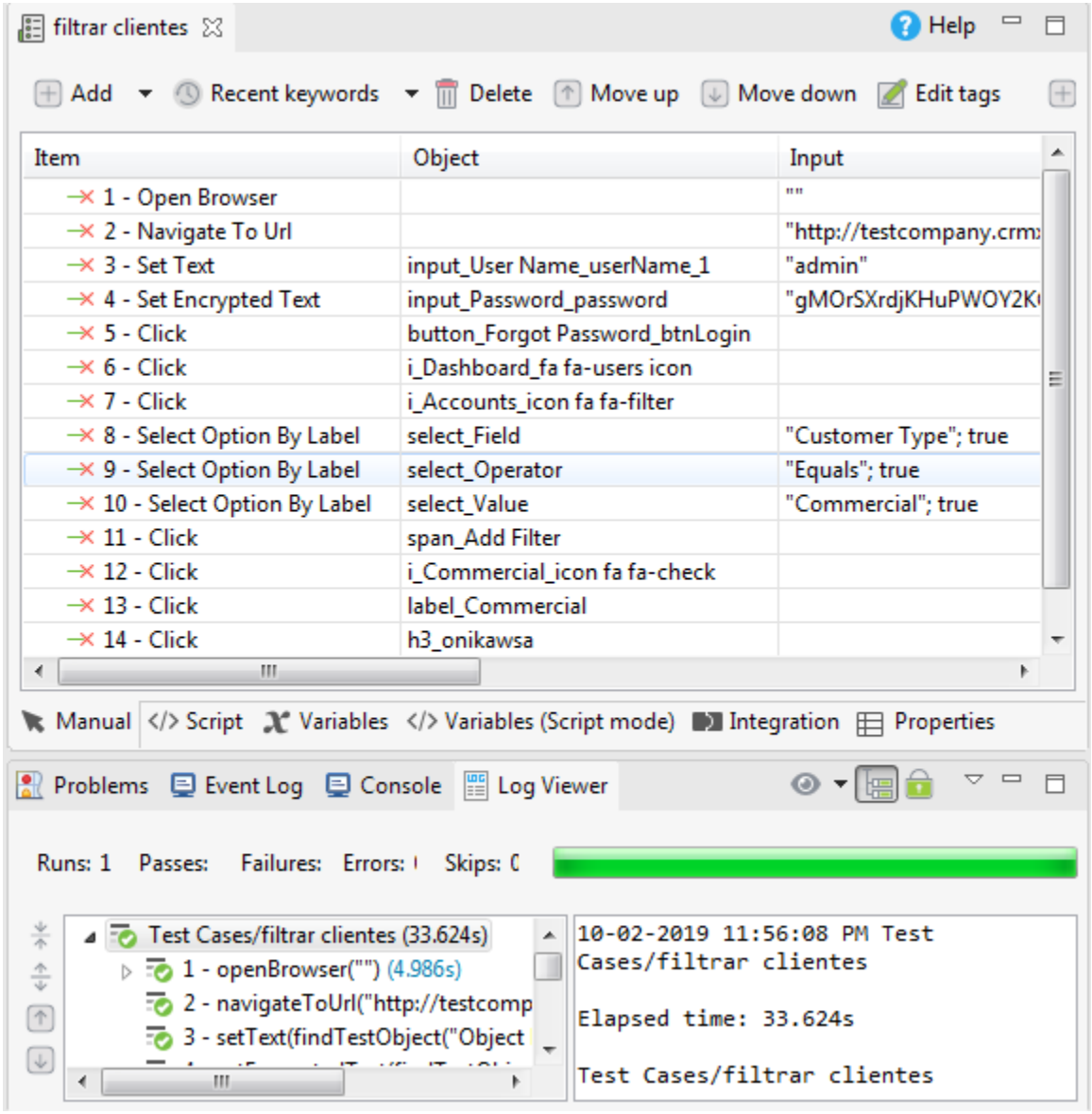

Ilustración 22 Caso de prueba BL007 - Filtrar Clientes

PB ID: BL005

Nombre: Detalle de Cuenta en Oportunidad

Descripción: Dirigirse a la vista de oportunidades ingresar a la oportunidad “New Opportunity” perteneciente a la cuenta “onikawsa” y seleccionar los detalles de la cuenta.

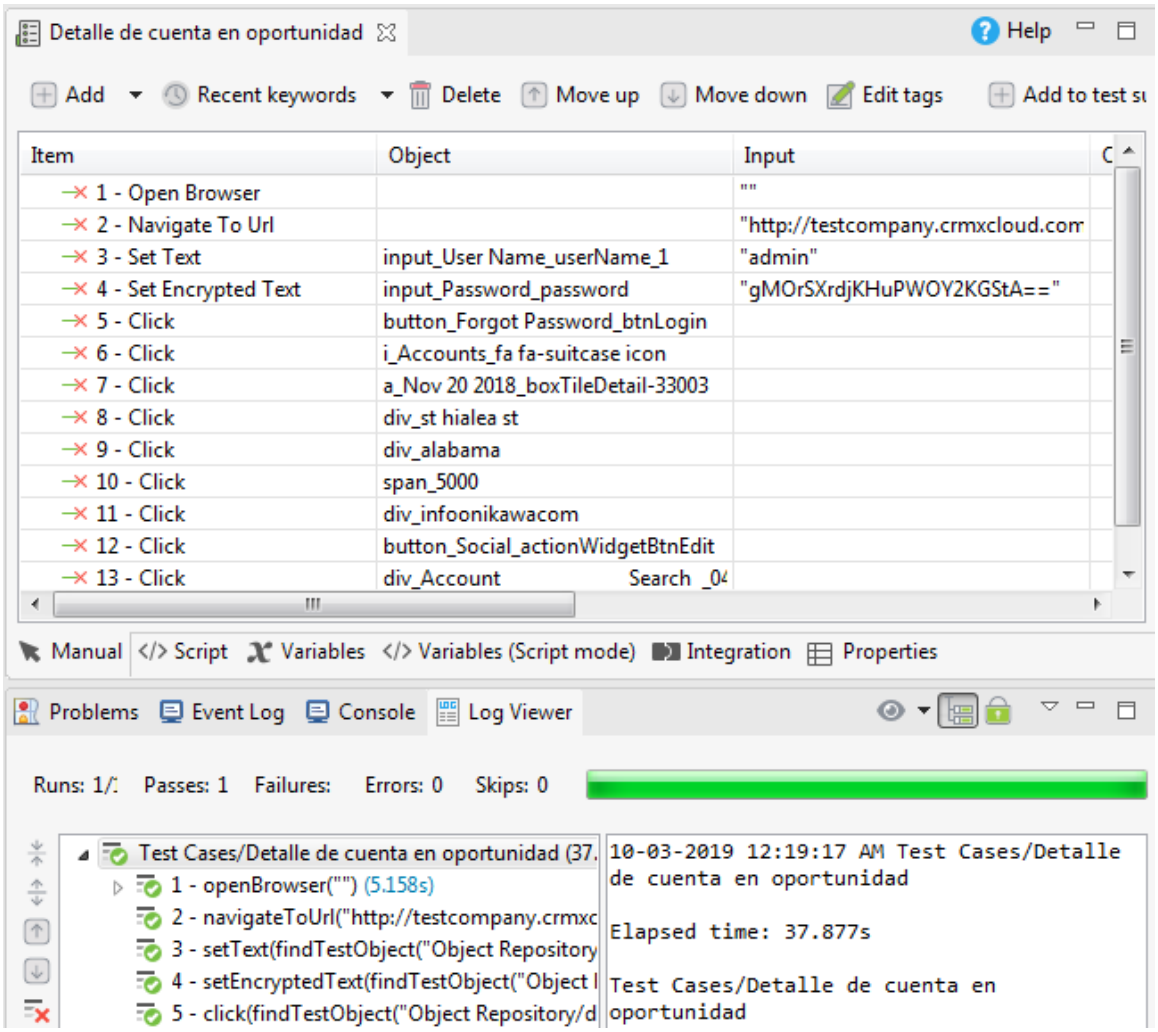

Ilustración 23 Caso de prueba BL005 - Detalle de Cuenta en Oportunidad

PB ID: BL004

Nombre: Listar Oportunidades por Cuenta

Descripción: Dirigirse a la vista de cuentas, seleccionar la cuenta “onikawsa”, en los widgets del extremo derecho localizar opción de llamar a oportunidades.

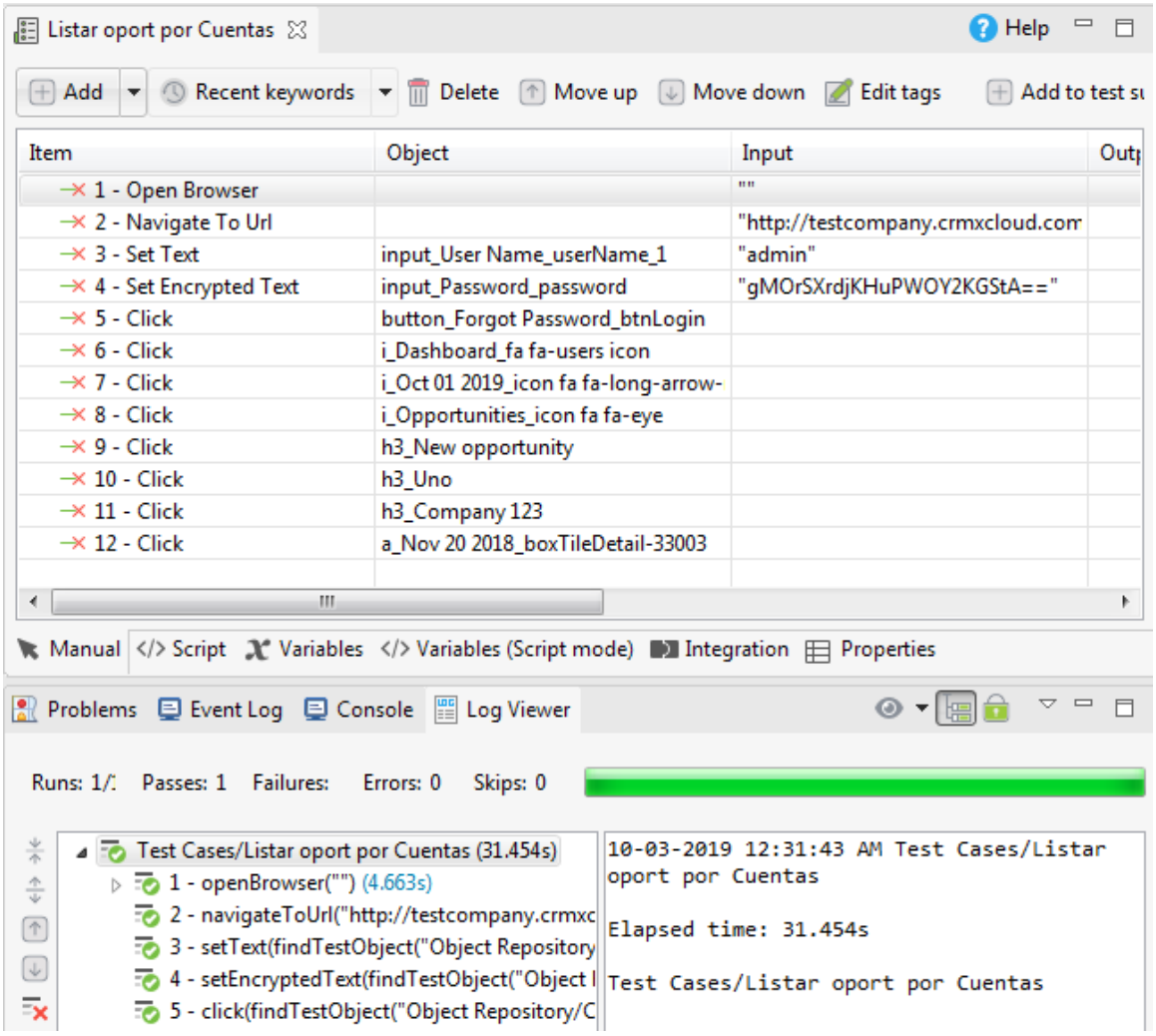

Ilustración 24 Caso de Prueba BL004 - Listar Oportunidades por Cuenta

Revisión del Sprint

Al concluir la semana 5 se citó a reunión al equipo Scrum donde también hizo presencia el encargado del área de informática de la empresa UBS para quien se está desarrollando el sistema. Luego de explicar los avances logrados en el primer Sprint y teniendo la aprobación de los representantes de la empresa se procede

a dar por terminada esta etapa y se procede a realizar la reunión de retrospectiva del Sprint.

5.2.4.5. Retrospectiva de cada Sprint

Resultados de retrospectiva del Sprint #1					
Tareas realizadas	7	**Sub-tareas creadas**		21	
Participantes	Equipo Scrum	***Semana***	Inicio semana 6	***Duración***	2 Horas
¿Qué hicimos bien?					
Se ha cumplido con los tiempos de entregas sin mayor percance.					
¿Qué mejora puede incorporar el equipo al proceso de forma inmediata?					
Distribuir de forma más equitativas las tareas asignadas a cada Sprint y administrar el número de nuevas solicitudes permitidas por parte del cliente a tareas en proceso.					
¿Qué vamos a hacer para conseguirlo?					
Indicar a Product Owner que en la planificación del Sprint determine si la empresa cliente desea agregar algún detalle a cada tarea y limitar el número de solicitudes durante proceso.					

Tabla 8 Retrospectiva de Sprint #1

5.2.5. Desarrollo del Sprint #2

5.2.5.1. Tareas y sub-tareas desarrolladas

SPRINT #2					
Semana de inicio #		**6**	**Semana de fin #**		**9**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
BL010	Multi empresas	Hecho	BLS026	Algoritmo de alta de Sub-dominio y DB individual por empresa automatizada	Hecho
			BLS027	Reenvió de dominio y validación según empresa	Hecho
			BLS028	Panel de creación de nueva empresa	Hecho
			BLS029	API de uso y validación de usuario	Hecho

SPRINT #2					
Semana de inicio #		**6**		**Semana de fin #**	**9**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
				por empresa	
BL013	Manejar archivos	Hecho	BLS033	Crear pantallas de archivos	Hecho
			BLS034	Crear tabla archivos y enlazar a controladores	Hecho
			BLS035	Codificación de crear, listar, eliminar y enlazar archivos con cuentas clientes.	Hecho
BL014	Tareas en cuentas de clientes	Hecho	BLS036	Diseñar y desarrollar pantalla de tareas	Hecho
			BLS037	Crear tabla de datos para tareas y enlazar a controladores	Hecho
			BLS038	Codificar algoritmos para crear, enlazar a clientes, editar, eliminar y listar tareas	Hecho
BL015	Detalle de cuenta cliente	Hecho	BLS039	Diseñar y desarrollar pantalla de detalles	Hecho
			BLS040	Enlazar a la pantalla todos los objetos relacionados al cliente (contactos, tareas, archivos)	Hecho
BL012	Notas de clientes	Hecho	BLS049	Agregar campos de notas a las pantallas detalles cuentas clientes	Hecho
			BLS050	Crear tabla de notas en la base de datos	Hecho
			BLS051	Codificar algoritmos para agregar, editar y eliminar notas	Hecho
BL016	Cambiar propietario de cuenta cliente	Hecho	BLS052	Agregar campo de propietario a pantalla edición de cliente	Hecho
			BLS053	Agregar campo Owner a tablas de clientes en la base de datos	Hecho
			BLS054	Agregar código de cambio de propietario al editar cliente.	Hecho

SPRINT #2					
Semana de inicio #		**6**	**Semana de fin #**		**9**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
BL018	Cambio de etapa de oportunidad	Hecho	BLS055	Agregar campo de cambio etapa oportunidad a pantalla detalle oportunidad	Hecho
			BLS056	Agregar tabla etapas de oportunidad a base de datos y enlazarla con tabla oportunidades	Hecho
			BLS057	Crear y codificar pantalla para agregar, eliminar, editar tipos de etapas	Hecho
			BLS058	Agregar código de cambio de etapas	Hecho

Tabla 9 Resultados Sprint #2

5.2.5.2. Base de datos SMA del Sprint

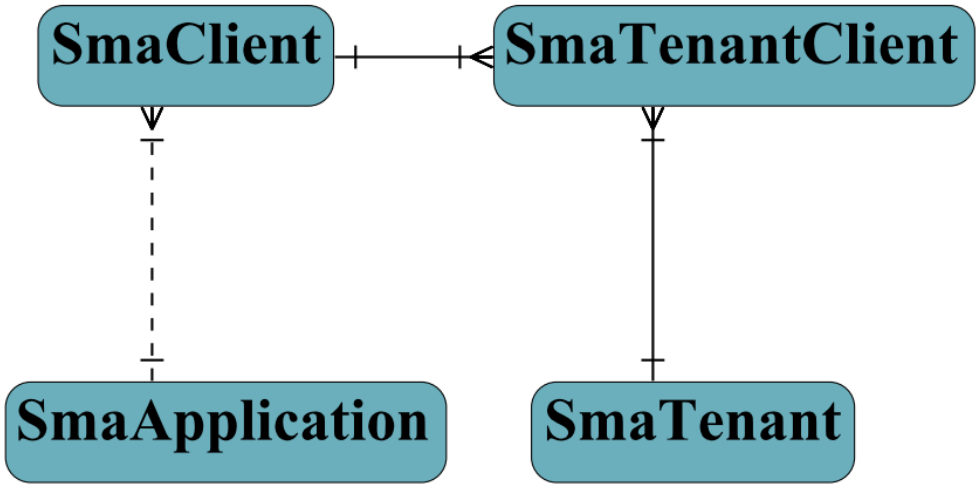

Ilustración 25 Tablas de datos SMA del Sprint #2

5.2.5.3. Base de datos CRM del Sprint

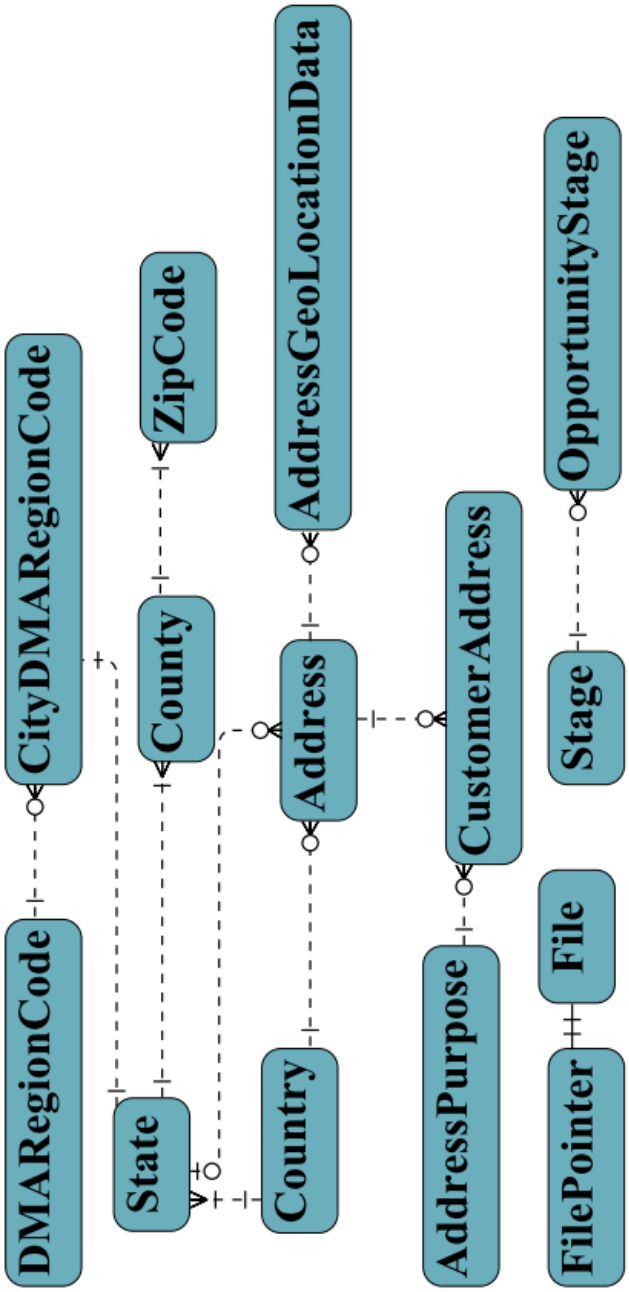

Ilustración 26 Tablas de datos del CRM del Sprint #2

5.2.5.4. Casos de Prueba del Sprint

PB ID: BL010

Nombre: Multiempresas

Descripción: Ingresar a dos subdominios distintos e iniciar sesión en cada uno de ellos y alternarse entre ambos para probar el funcionamiento de la estructura Multi-tenant.

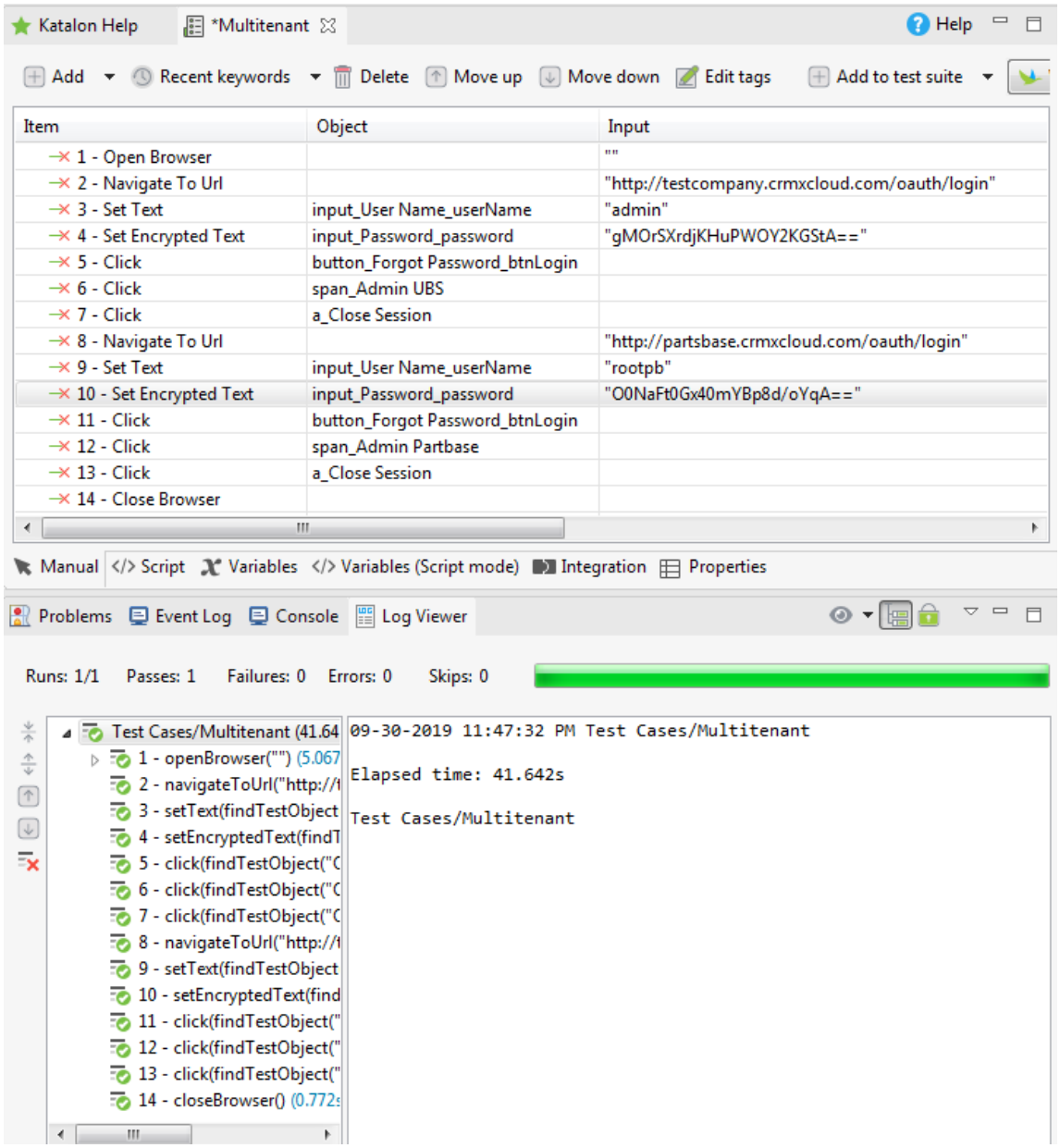

Ilustración 27 Caso de Pruebas BL010 - Multiempresas

PB ID: BL013

Nombre: Manejar Archivos

Descripción: Dirigirse con ayuda del menú a la sección archivos, seleccionar un elemento y proceder a eliminarlo.

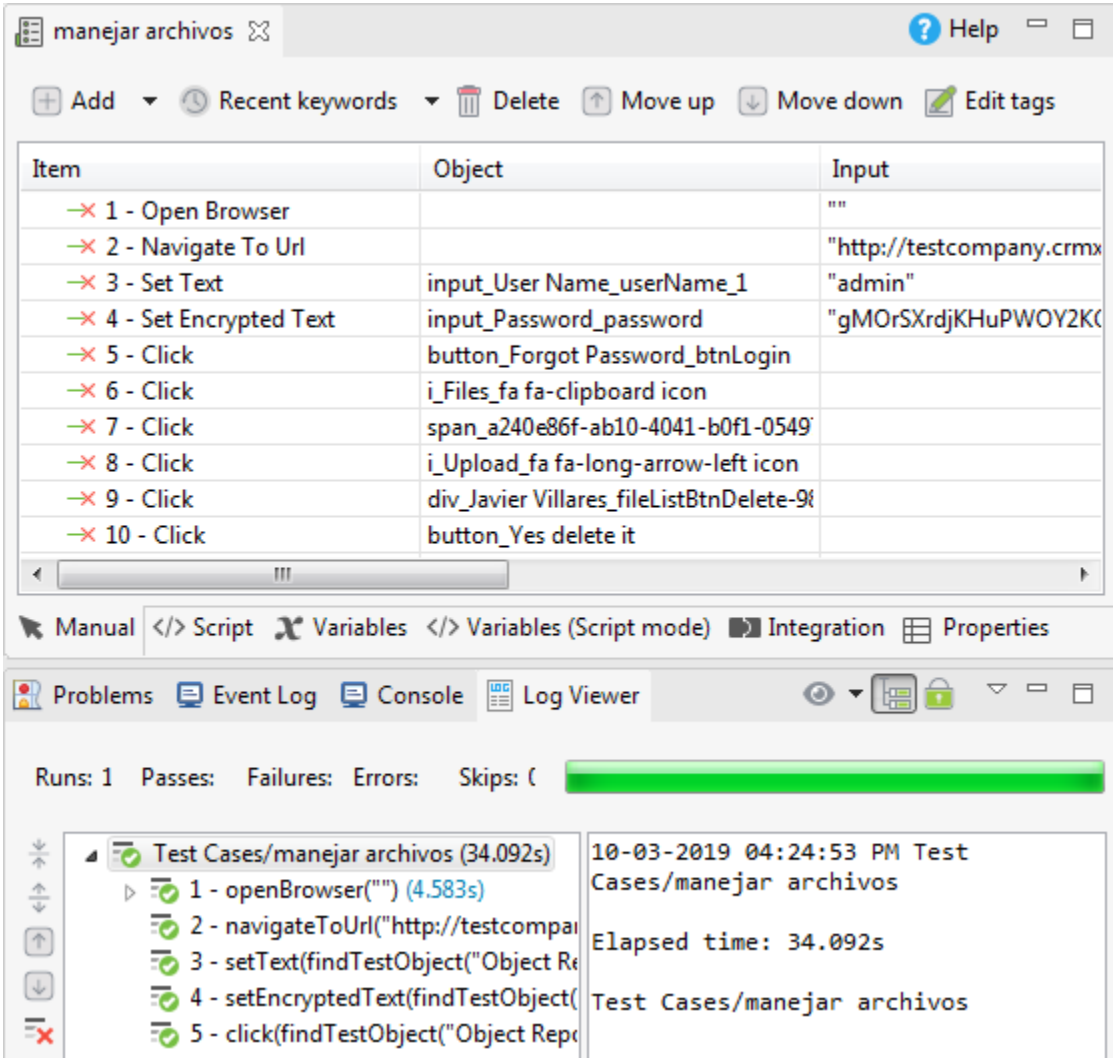

Ilustración 28 Caso de prueba BL013 - Manejar Archivos

PB ID: BL014

Nombre: Filtrar Clientes

Descripción: Dirigirse a la vista de todos los clientes luego utilizar los filtros para buscar clientes de tipo comercial.

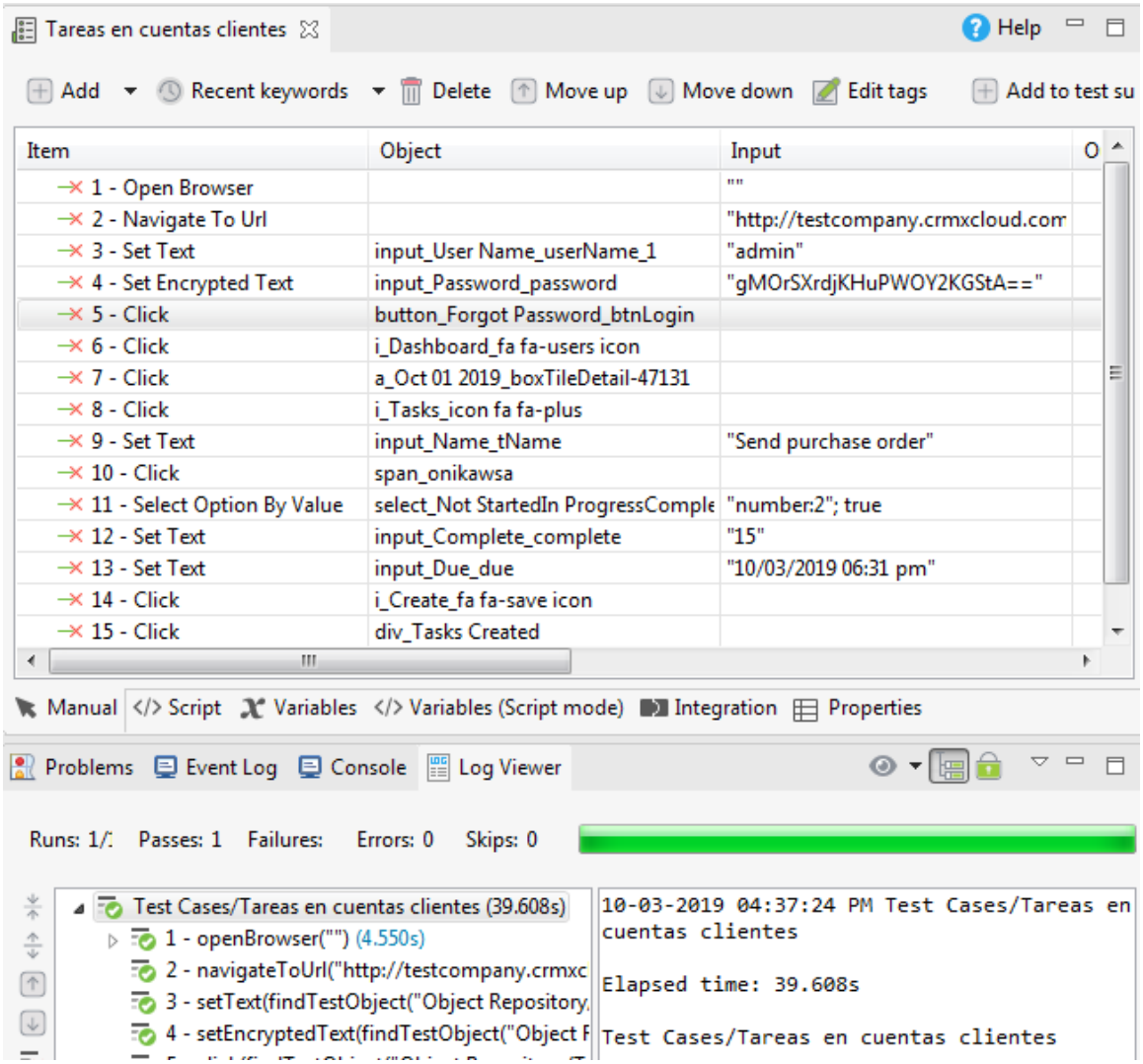

Ilustración 29 Caso de prueba BL014 - Tareas en cuentas clientes

PB ID: BL016

Nombre: Cambiar Propietario de Cuenta

Descripción: Seleccionar una cuenta, mostrar los detalles, editarla y asignar un nuevo “Propietario” de la misma luego guardar los cambios.

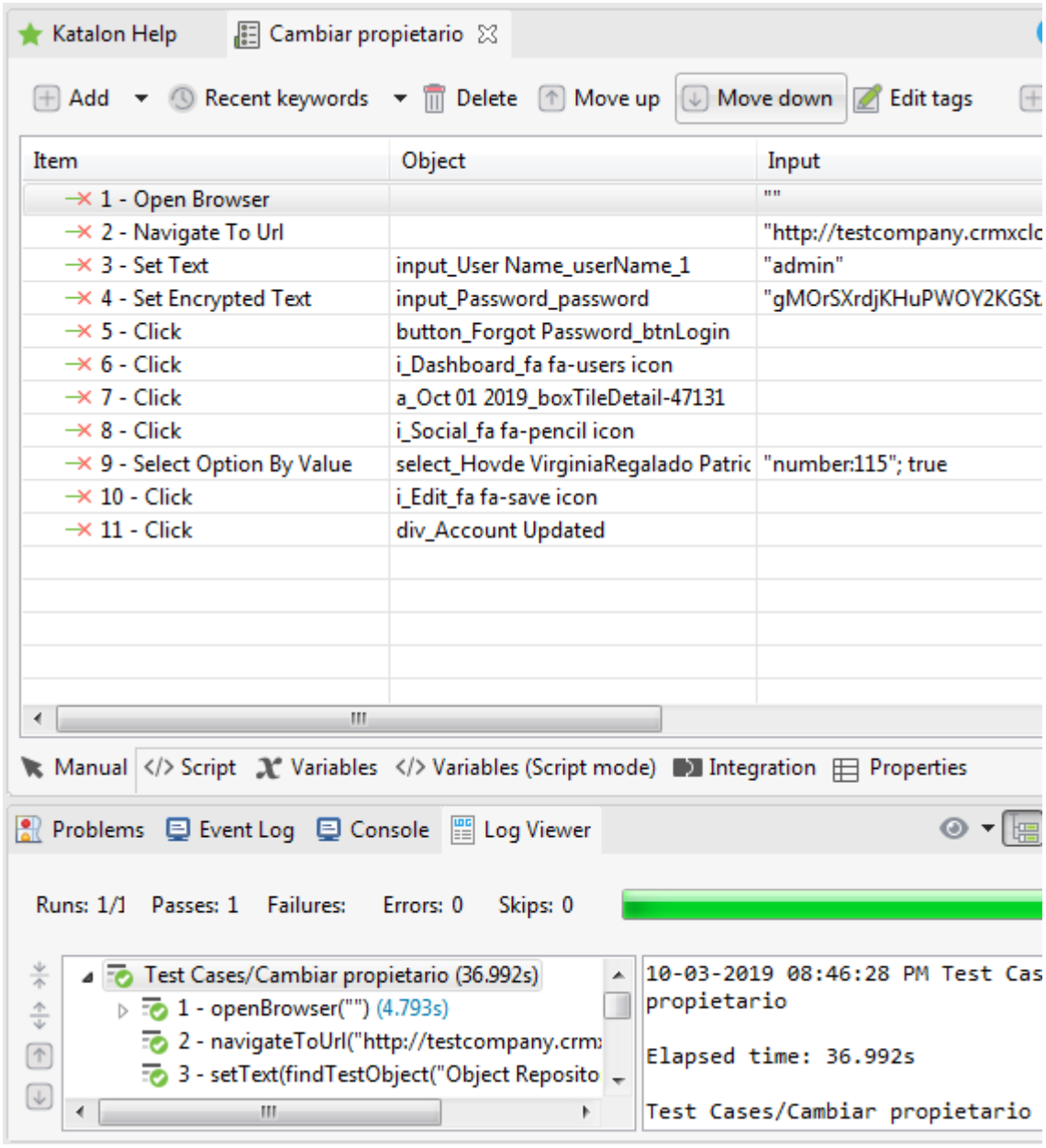

Ilustración 30 Caso de prueba BL016 - Cambiar Propietario de Cuenta

PB ID: BL018

Nombre: Cambiar Etapa de Oportunidad

Descripción: Seleccionar una oportunidad, desde la vista detallada seleccionar uno de los estados en secuencia, luego editar la oportunidad y realizar el mismo intento, terminar al guardar la oportunidad.

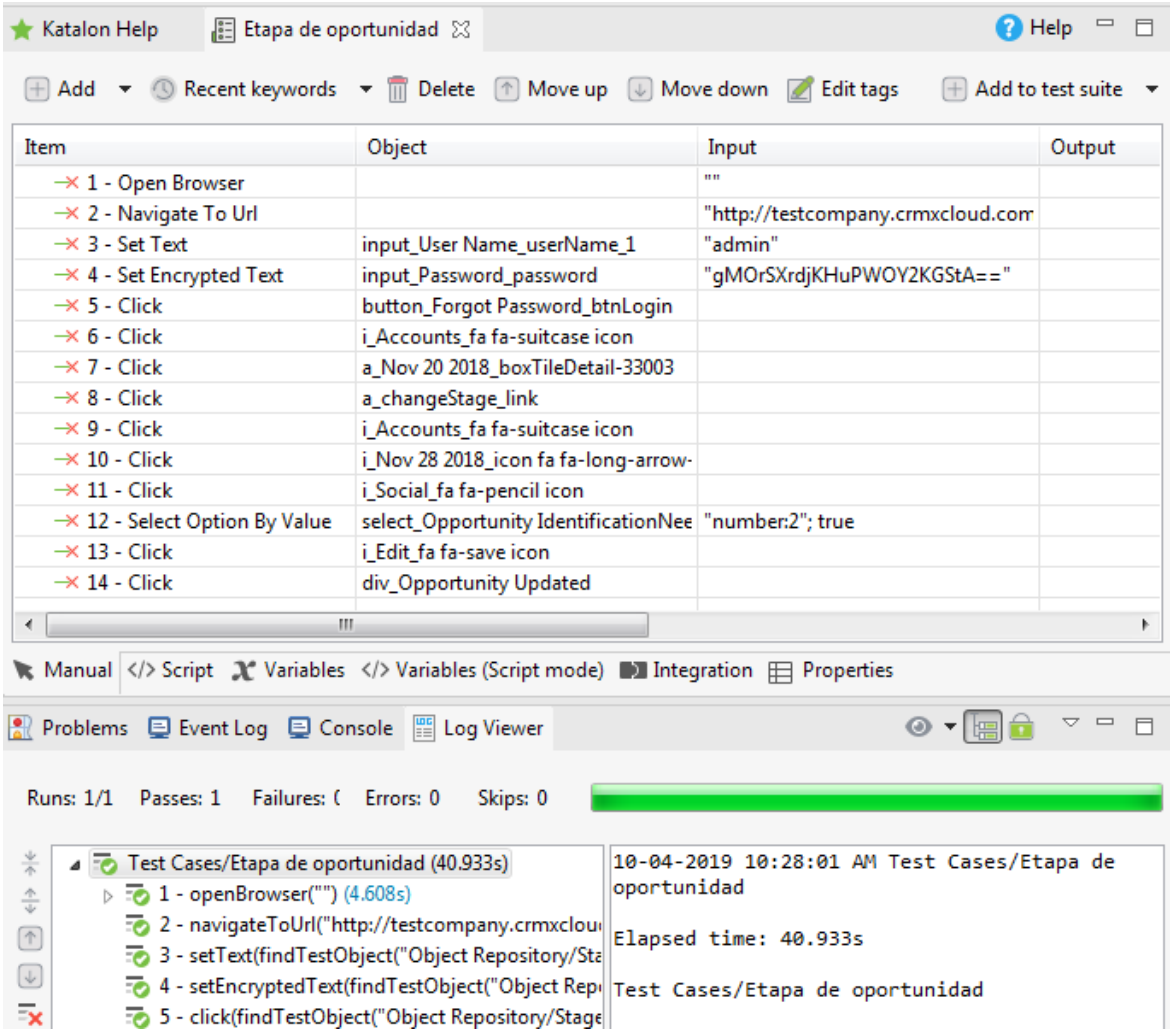

Ilustración 31 Caso de prueba BL018 - Cambiar Etapa de Oportunidad

5.2.5.5. Revisión del Sprint

Al concluir la semana 8 se citó a reunión al equipo Scrum, luego de explicar el retraso de entrega de avances por motivos externos a ámbitos laborales se procedió a demostrar los avances logrados al Product Owner el cual acepto con suma satisfacción el desarrollo entregado.

5.2.5.6. Retrospectiva del Sprint

Resultados de retrospectiva del Sprint #2					
Tareas realizadas	11	**Sub-tareas creadas**		33	
Participantes	Equipo Scrum	***Semana***	Inicio semana 9	***Duración***	2 Horas
¿Qué hicimos bien?					
¿Qué mejora puede incorporar el equipo al proceso de forma inmediata?					
Documentar cada nuevo módulo que se ingresa al sistema y realizar respectivos comentarios a la codificación en especial a lo relacionado con la arquitectura multi-tenant.					
¿Qué vamos a hacer para conseguirlo?					
Project Manager se compromete a dar seguimiento a los documentos y realizar Pull Request al repositorio Git para verificar si se van modificando conforme avanza el Sprint.					

Tabla 10 Retrospectiva de Sprint #2

5.2.6. Desarrollo del Sprint #3

5.2.6.1. Tareas y sub-tareas desarrolladas

SPRINT #3					
Semana de inicio #		**10**		**Semana de fin #**	**14**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
BL017	Cuentas Leads	Hecho	BLS059	Diseñar y codificar pantallas Leads	Hecho
			BLS060	Crear tablas y relaciones en la Base de Datos para Leads	Hecho
			BLS061	Codificación para crear, editar, mostrar, detallar y eliminar Leads	Hecho
BL019	Actualizar tipo de cuenta Lead/Oportunidad/ Cuenta cerrada	Hecho	BLS071	Agregar botón de cambio de tipo de cuenta	Hecho
			BLS072	Codificar algoritmo de cambio de estado de la cuenta cliente	Hecho
BL024	Calendario de citas	Hecho	BLS077	Crear tabla de datos para calendario	Hecho
			BLS078	Crear y desarrollar vista de calendario	Hecho
			BLS079	Agregar, editar, eliminar, actualizar actividades de calendario	Hecho
			BLS080	Agregar características calendarios a pantallas de detalles en cuentas, oportunidades y leads	Hecho
BL021	Oportunidad ganada/perdida	Hecho	BLS062	Agregar columna a tabla oportunidades	Hecho
			BLS063	Codificar controlador para almacenar y actualizar	Hecho

SPRINT #3					
Semana de inicio #		**10**		**Semana de fin #**	**14**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
				estado	
			BLS064	Agregar opción al diseño web	Hecho
BL026	Listar contactos en cuentas cliente	Hecho	BLS065	Filtrar contactos por cuentas	Hecho
			BLS066	Mostrar lista de contactos en pantalla cuentas	Hecho
BL027	Listado de tarea en cuentas cliente	Hecho	BLS067	Filtrar tareas por cuentas	Hecho
			BLS068	Mostrar lista de tareas en pantalla cuentas	Hecho
BL032	Territorios	Hecho	BLS069	Crear tablas territorios	Hecho
			BLS070	Diseñar y desarrollar plantillas de territorios	Hecho

Tabla 11 Resultados Sprint #3

5.2.6.2. Base de datos CRM del Sprint

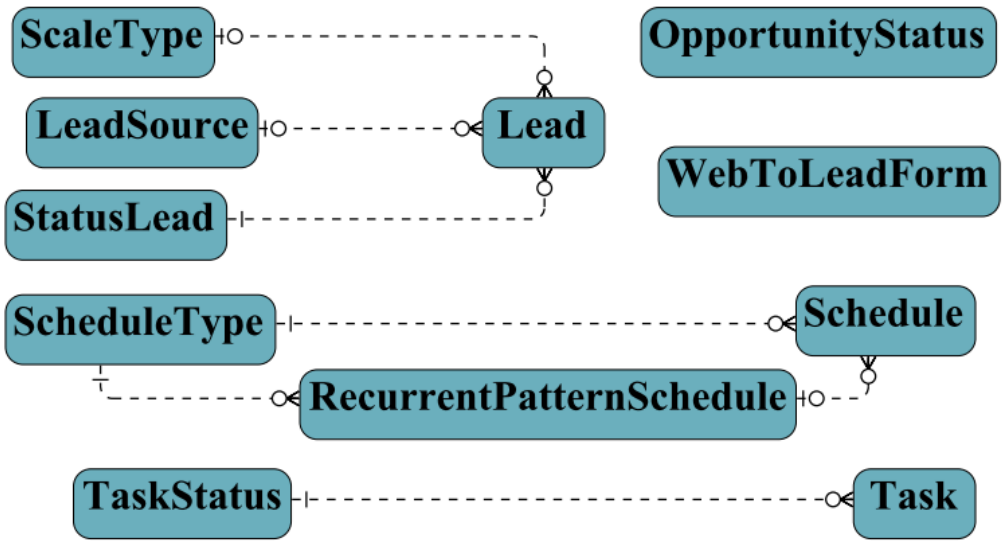

Ilustración 32 Tablas de datos del CRM del Sprint #3

5.2.6.3. Casos de Prueba del Sprint

PB ID: BL019

Nombre: Actualizar tipo de cuenta Lead/Oportunidad/Cuenta cerrada

Descripción: Seleccionar un cliente que está marcado como Cuenta pasar su estado a Oportunidad luego llevarlo a Leads y retornar a través de Oportunidad hasta Cuenta nuevamente.

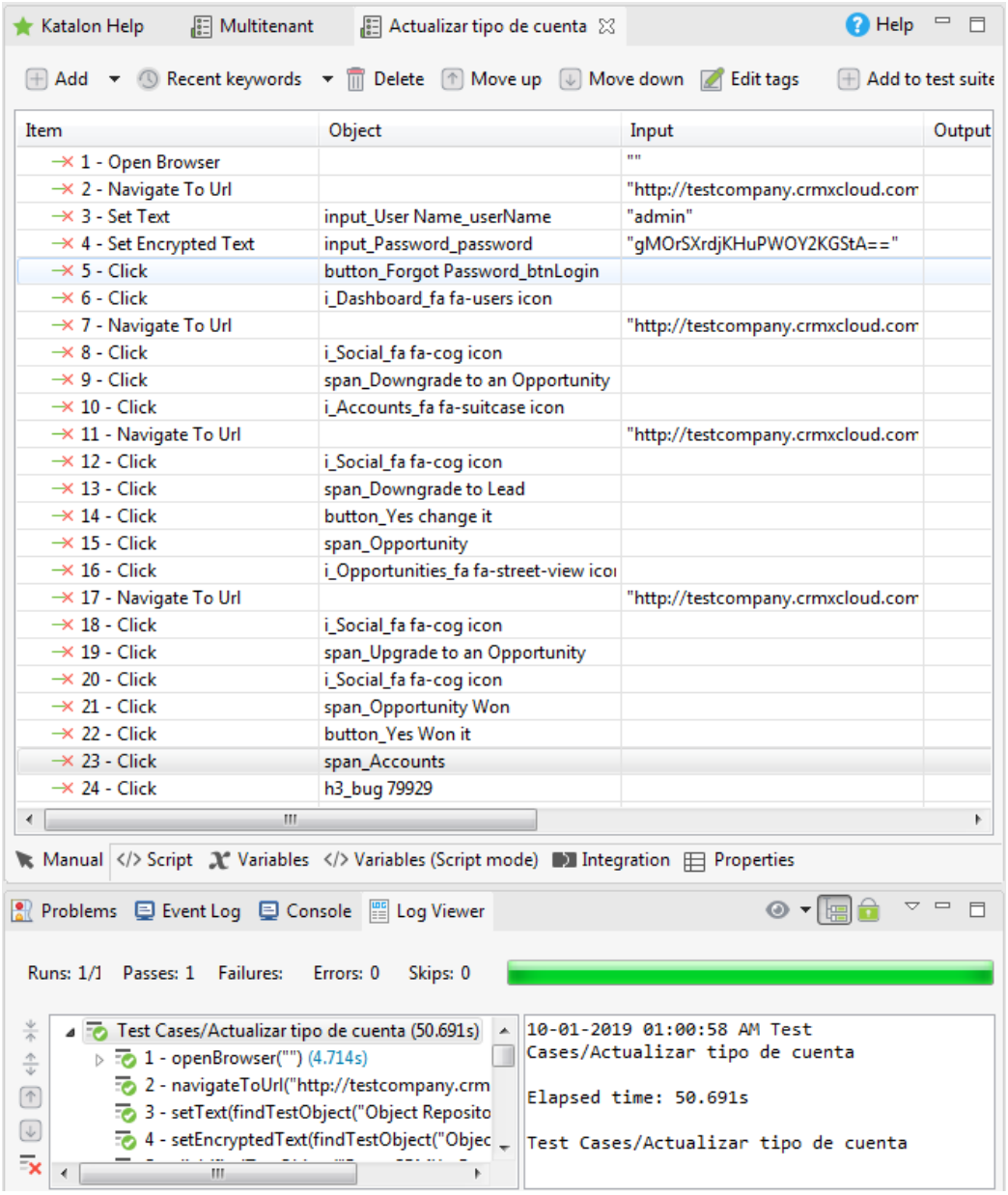

Ilustración 33 Caso de Prueba BL019 - Actualizar Estado de cliente

PB ID: BL017

Nombre: Leads

Descripción: Dirigirse a la sección Leads, crear un nuevo cliente prospecto, editar el mismo cliente y guardar sin cambios luego, por último, eliminar el prospecto.

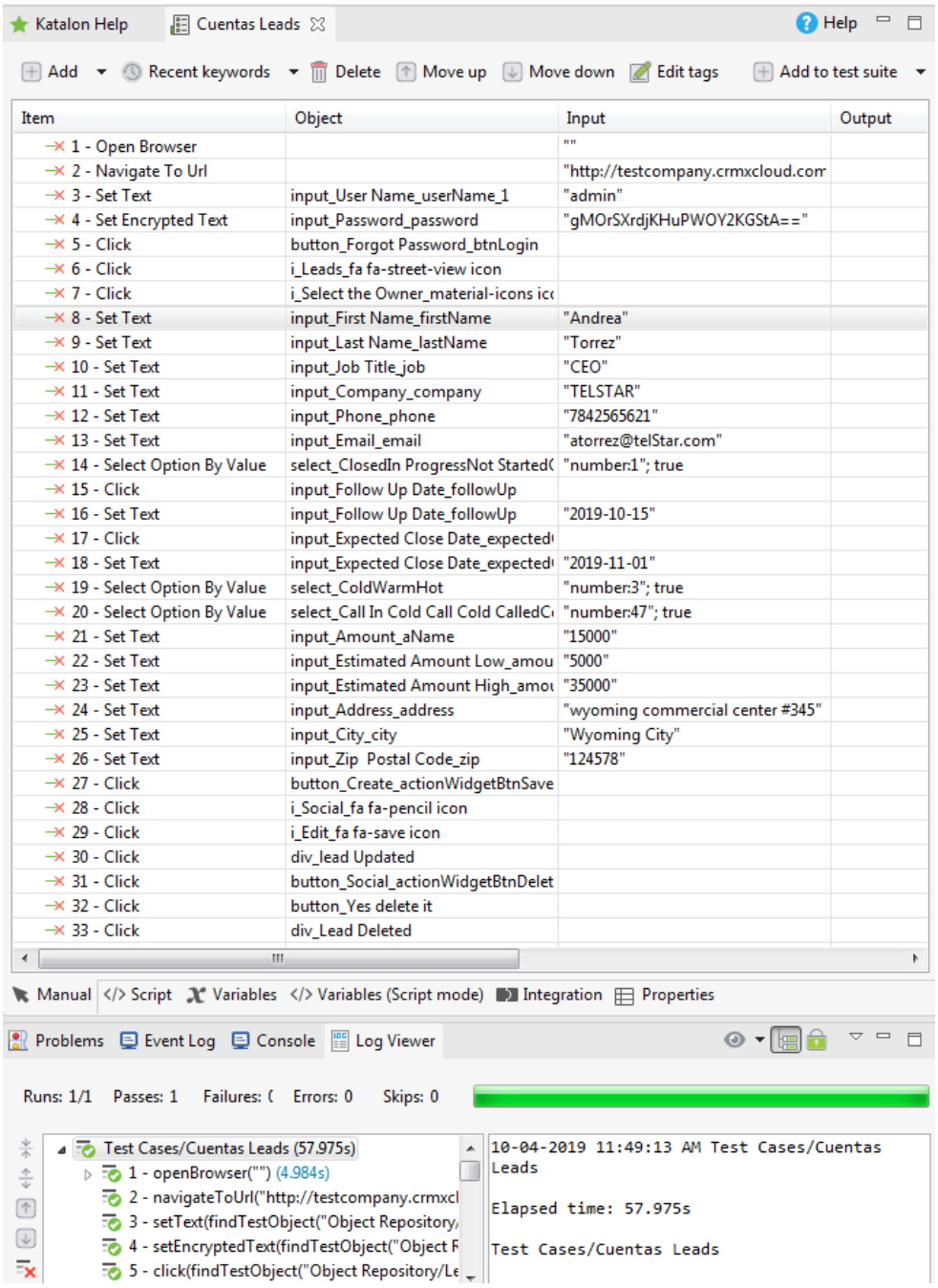

Ilustración 34 Caso de prueba BL017 - Leads

PB ID: BL024

Nombre: Calendario de Citas

Descripción: Dirigirse a la sección “Schedule”, crear un nuevo evento en el calendario con un cliente y guardarlo.

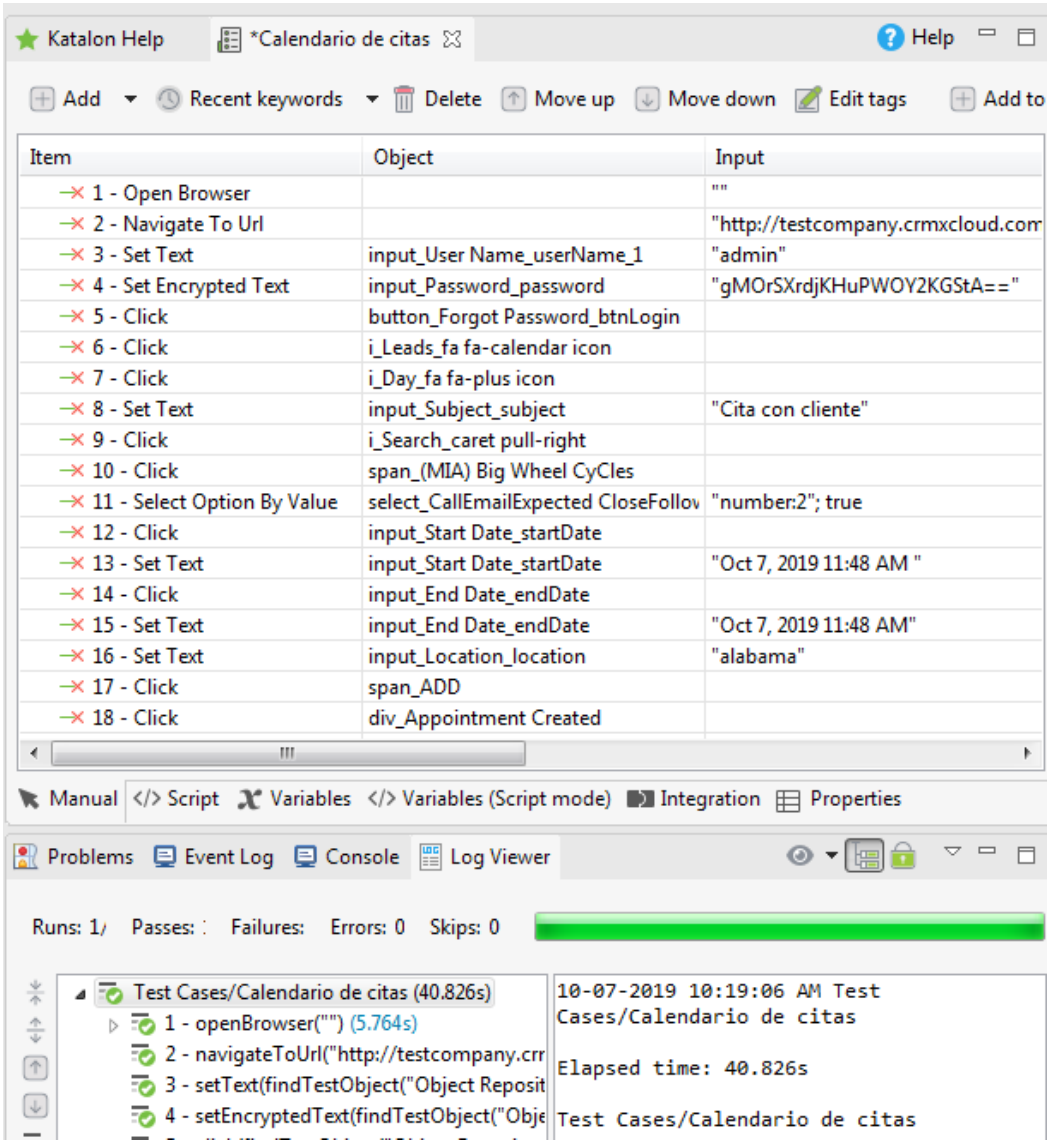

Ilustración 35 Caso de prueba BL024 - Calendario de Citas

PB ID: BL021

Nombre: Oportunidad ganada/perdida

Descripción: Dirigirse a la sección “Opportunity”, mostrar los detalles de la oportunidad y asignar el estado de oportunidad Ganada, aceptar y luego seleccionar otra oportunidad para asignarle el estado de oportunidad perdida.

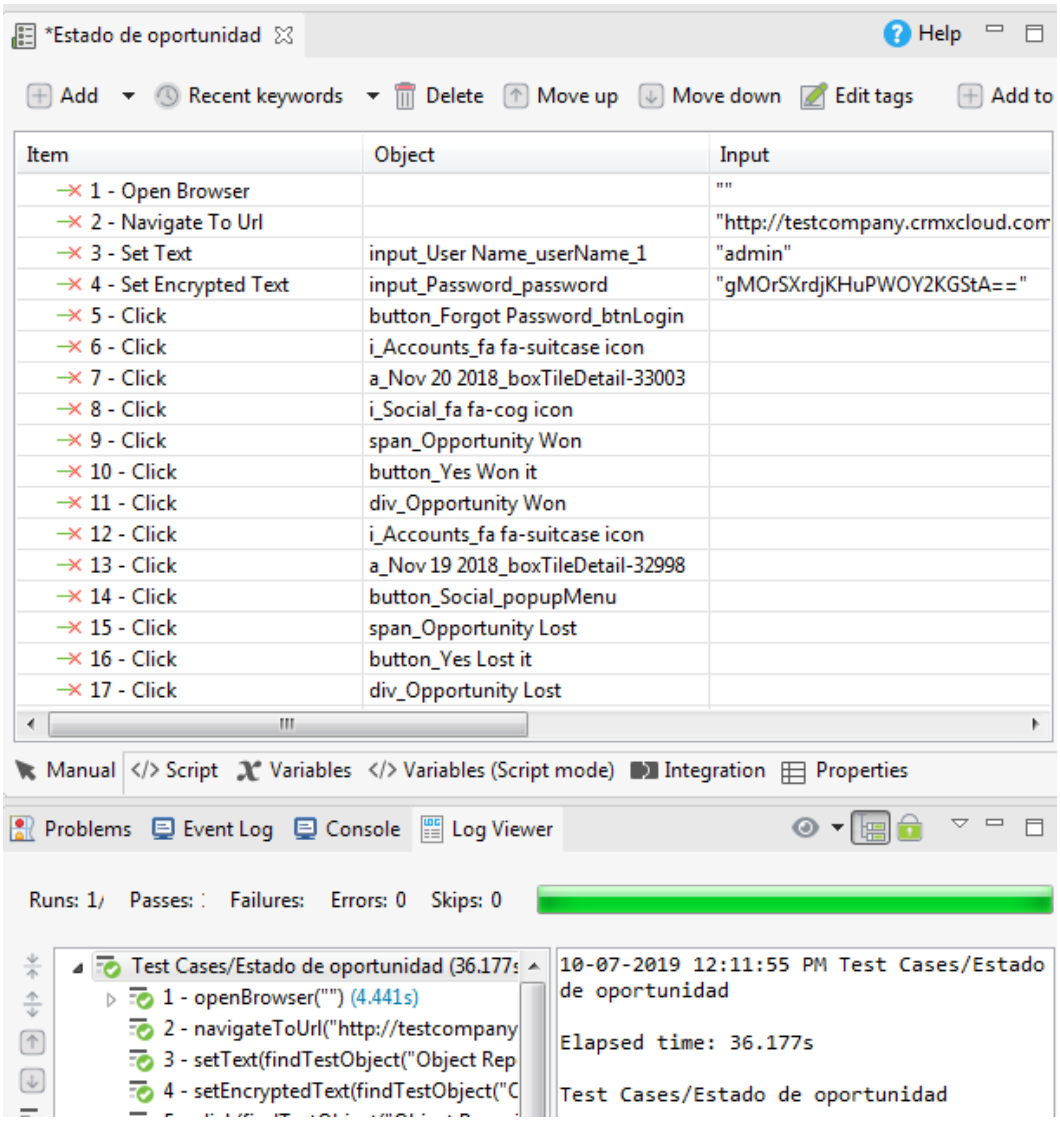

Ilustración 36 Caso de prueba BL021 - Oportunidad ganada/perdida

PB ID: BL026

Nombre: Listar Contactos en Cuentas Cliente

Descripción: Dirigirse a la sección Opportunity, Account o Leads, seleccionar uno de los clientes y de los widgets de la derecha inferior visualizar los contactos asignados a este, terminar agregando un nuevo contacto.

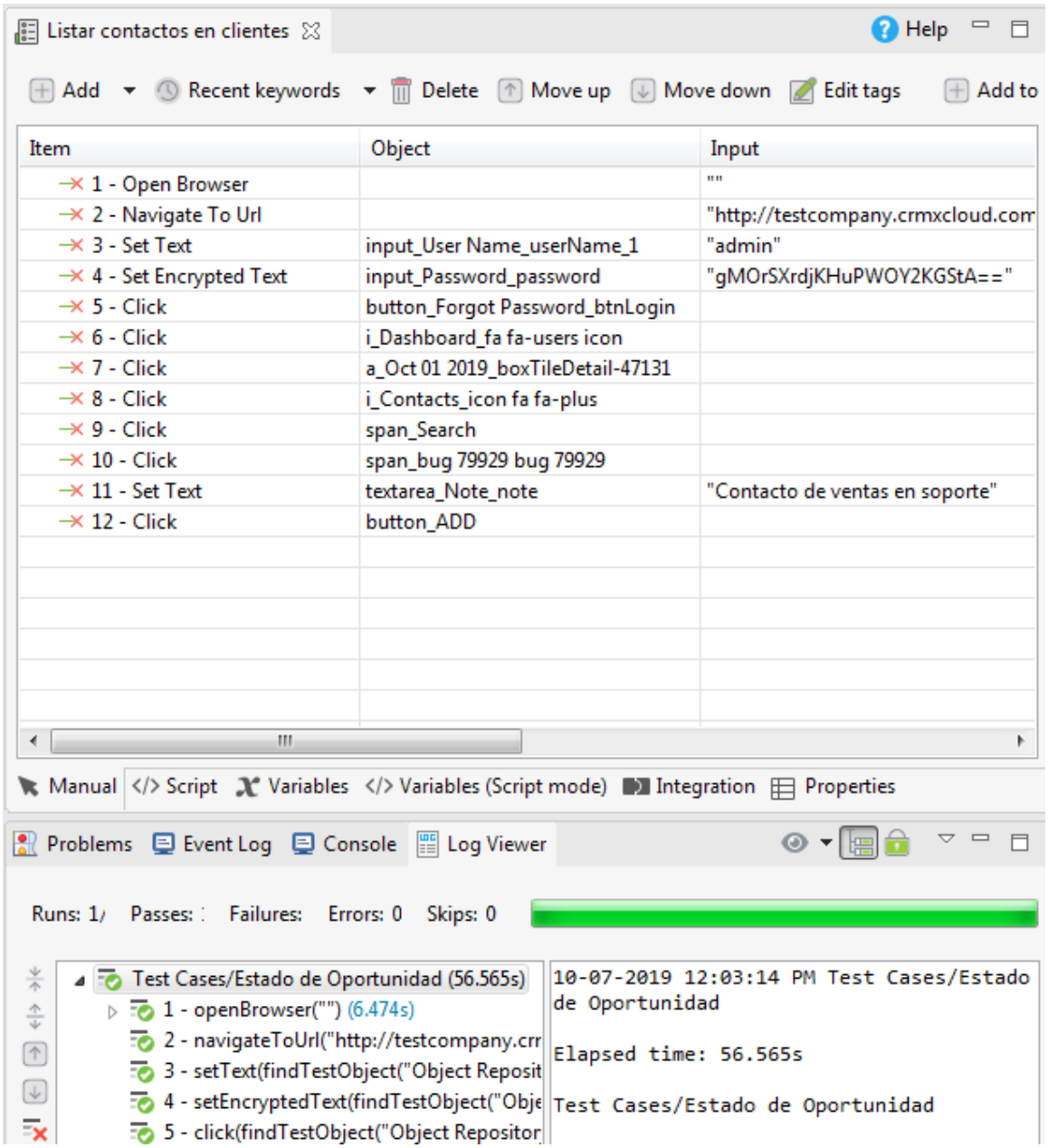

Ilustración 37 Caso de prueba BL026 - Listar Contactos en Cuentas Cliente

PB ID: BL027

Nombre: Listar Tareas en Cuentas Cliente

Descripción: Dirigirse a la sección Opportunity, Account o Leads, seleccionar uno de los clientes y de los widgets de la derecha inferior visualizar las tareas asignados a este.

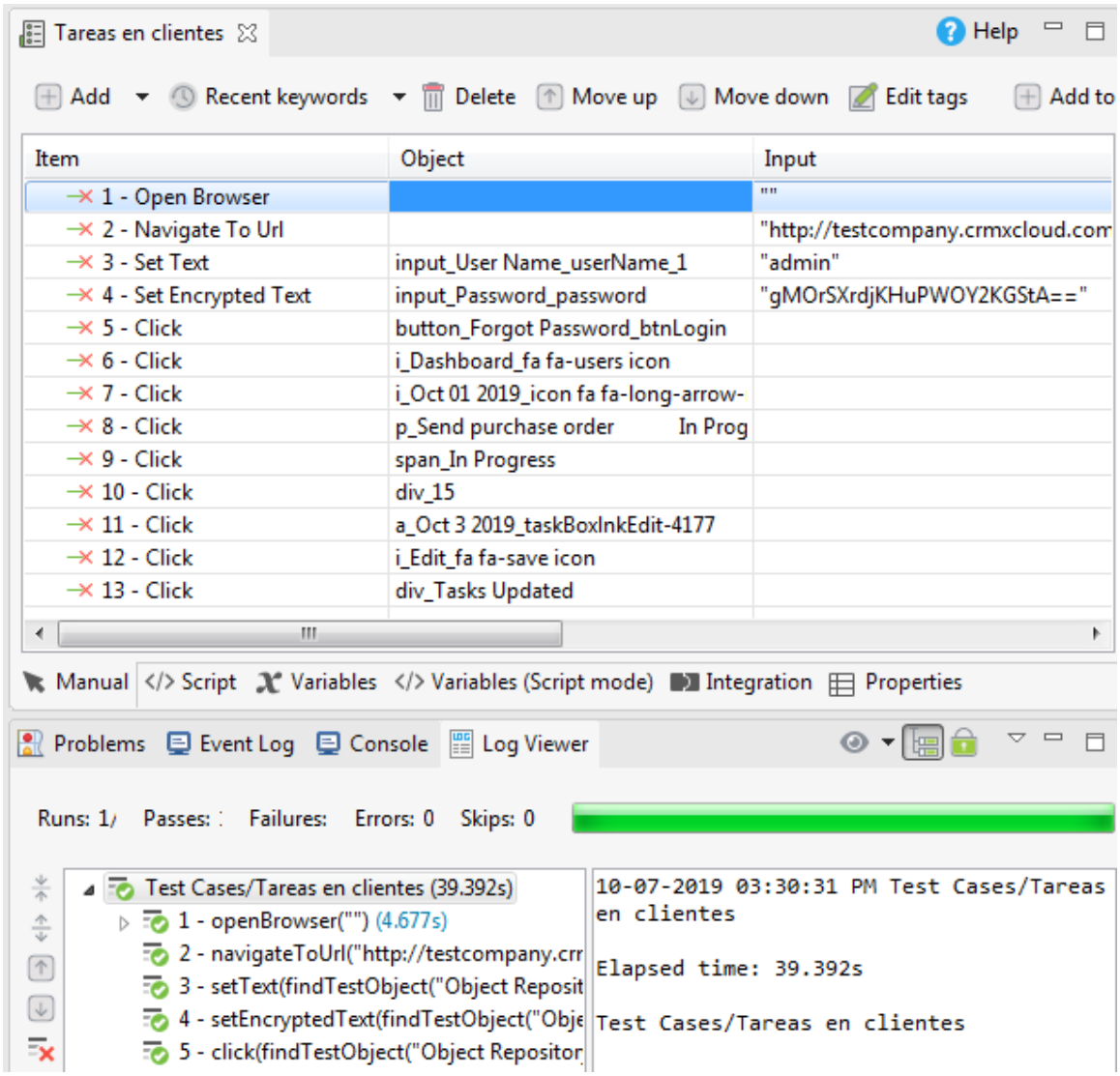

Ilustración 38 Caso de prueba BL027 - Listar Tareas en Cuentas Cliente

PB ID: BL032

Nombre: Territorios

Descripción: Dirigirse a la sección territorio y agregar un nuevo territorio de trabajo.

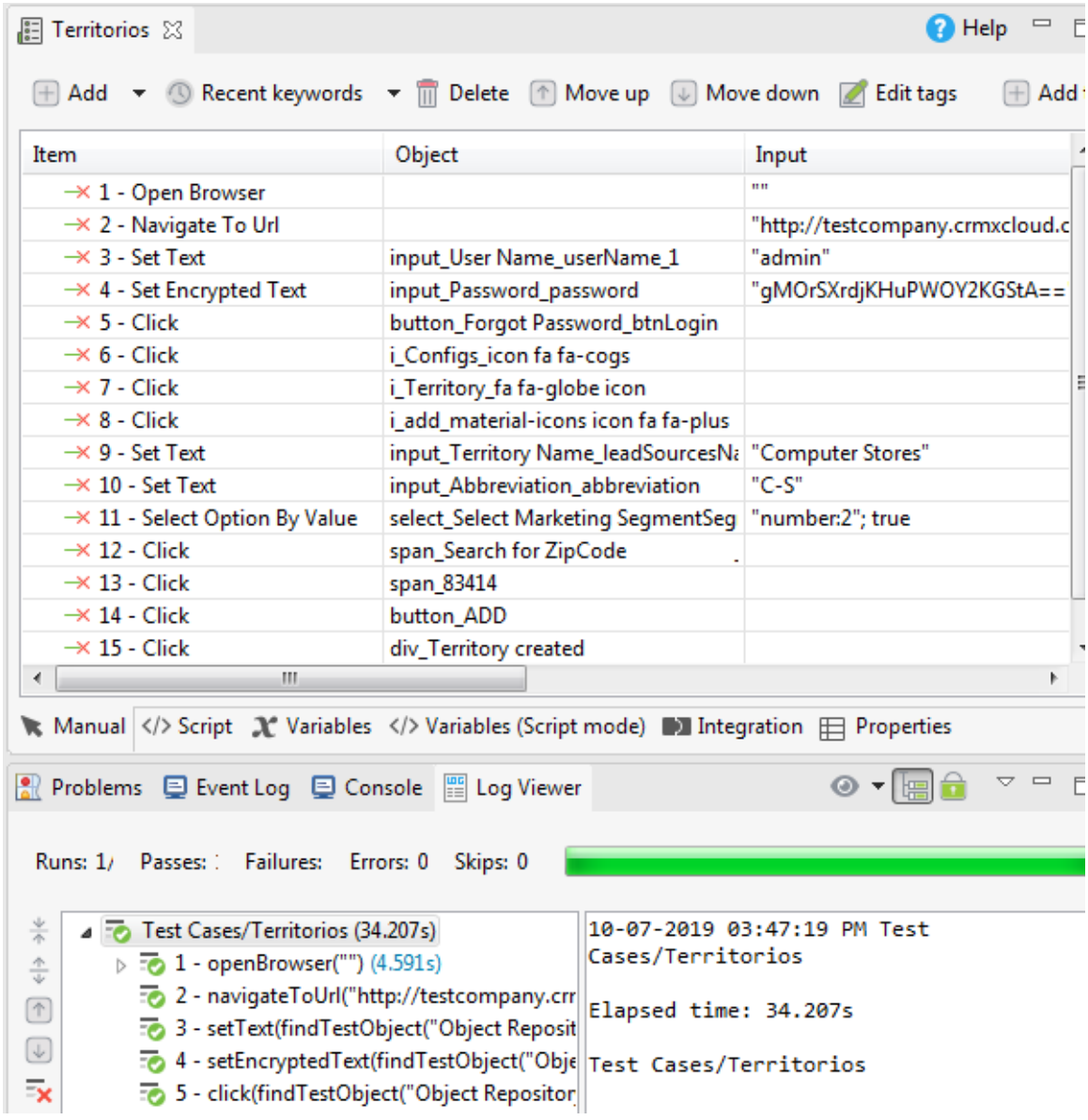

Ilustración 39 Caso de prueba Territorios

5.2.6.4. Revisión del Sprint

Al concluir las tareas del presente Sprint se citó a reunión al equipo Scrum, se han entregado los avances correspondientes, en este caso el Product Owner solo nos ha podido acompañar vía video llamada a lo cual sin mayor problema se aceptaron los avances para poder continuar con el proceso de desarrollo.

5.2.6.5. Retrospectiva del Sprint

Resultados de retrospectiva del Sprint #3					
Tareas realizadas	8	**Sub-tareas creadas**		28	
Participantes	Equipo Scrum	***Semana***	Inicio semana 15	***Duración***	2 Horas
¿Qué hicimos bien?					
Logramos re-factorizar código perteneciente a Sprints anteriores.					
¿Qué mejora puede incorporar el equipo al proceso de forma inmediata?					
Solicitar una mejora en el tiempo de entrega del área de diseño gráfico, demoramos tiempo esperando a que esté listo el arte en que debemos basarnos para poder desarrollar las tareas.					
¿Qué vamos a hacer para conseguirlo?					
Project Owner se compromete a dar seguimiento al equipo de diseño gráfico para no retrasar el proyecto, se compromete a tener todos los cambios solicitados antes de iniciar próximo Sprint.					

Tabla 12 Retrospectiva de Sprint #3

5.2.7. Desarrollo del Sprint #4

5.2.7.1. Tareas y sub-tareas desarrolladas

SPRINT #4					
Semana de inicio #		**15**	**Semana de fin #**		**18**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
BL031	Reportes	Hecho	BLS081	Diseñar y desarrollar plantillas de reportes	Hecho
			BLS082	Integrar plantillas al sistema, sección reportes	Hecho
			BLS083	Codificación de filtros y enlaces a base de datos	Hecho
BL011	Campos personalizados	Hecho	BLS084	Diseñar y desarrollar pantalla de campos personalizados	Hecho
			BLS085	Crear tabla de datos para campos personalizados	Hecho
			BLS086	Codificación para agregar, editar, eliminar campos personalizados de tipo múltiple	Hecho
BL022	Panel de Administración	Hecho	BLS087	Diseñar y desarrollar pantalla de panel de administración	Hecho
			BLS088	Crear tablas de datos de configuraciones y enlazar a controlador	Hecho
			BLS089	Codificar edición de ajustes	Hecho
BL029	Roles de usuarios	Hecho	BLS096	Crear tablas de roles y permisos	Hecho
			BLS097	Codificar pantallas para asignar roles y permisos	Hecho
			BLS098	Codificar algoritmos de distribución de accesos por usuarios según sus permisos	Hecho
BL033	Departamentos y áreas laborales	Hecho	BLS090	Crear tablas departamentos y áreas laborales	Hecho
			BLS091	Diseñar y desarrollar plantillas de	Hecho

SPRINT #4					
Semana de inicio #		**15**		**Semana de fin #**	**18**
Tareas y sub-tareas desarrolladas					
PB ID	**Nombre**	**Estado**	**Sub-Tareas**		**Estado**
			PB ID	**Descripción**	
				áreas laborales y departamentos	
			BLS092	Enlazar a módulo de empleados	Hecho
BL034	Listado de empleados	Hecho	BLS093	Crear tabla en base de datos para empleados	Hecho
			BLS094	Diseñar y desarrollar plantillas web para empleados	Hecho
			BLS095	Codificar eliminar, agregar, editar y actualizar empleados.	Hecho

Tabla 13 Resultados Sprint #4

5.2.7.2. Base de datos SMA del Sprint

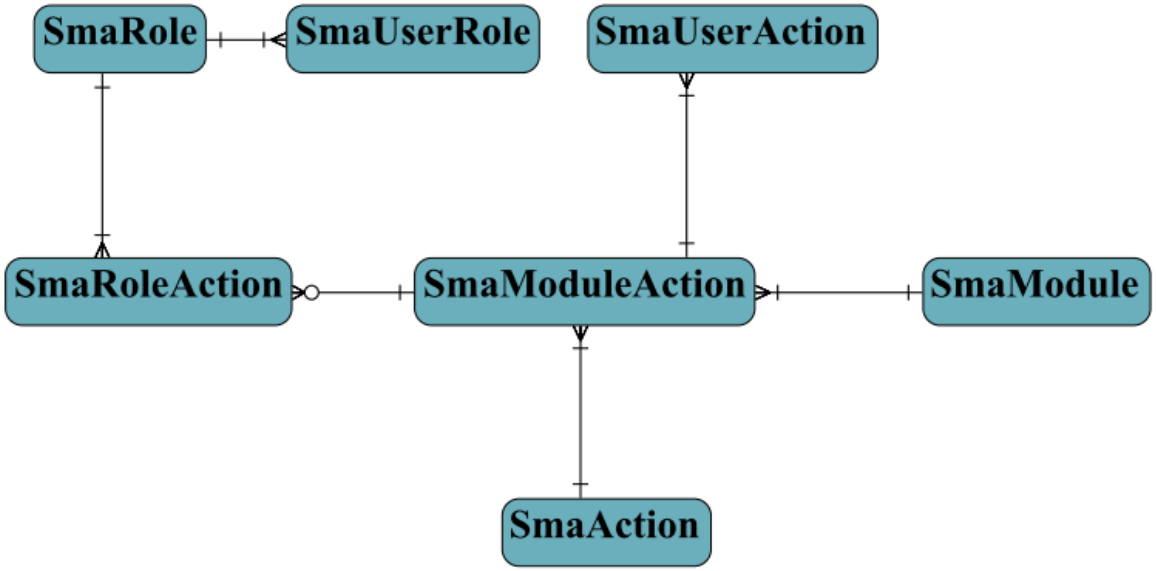

Ilustración 40 Tablas de datos SMA del Sprint #4

5.2.7.3. Base de datos CRM del Sprint

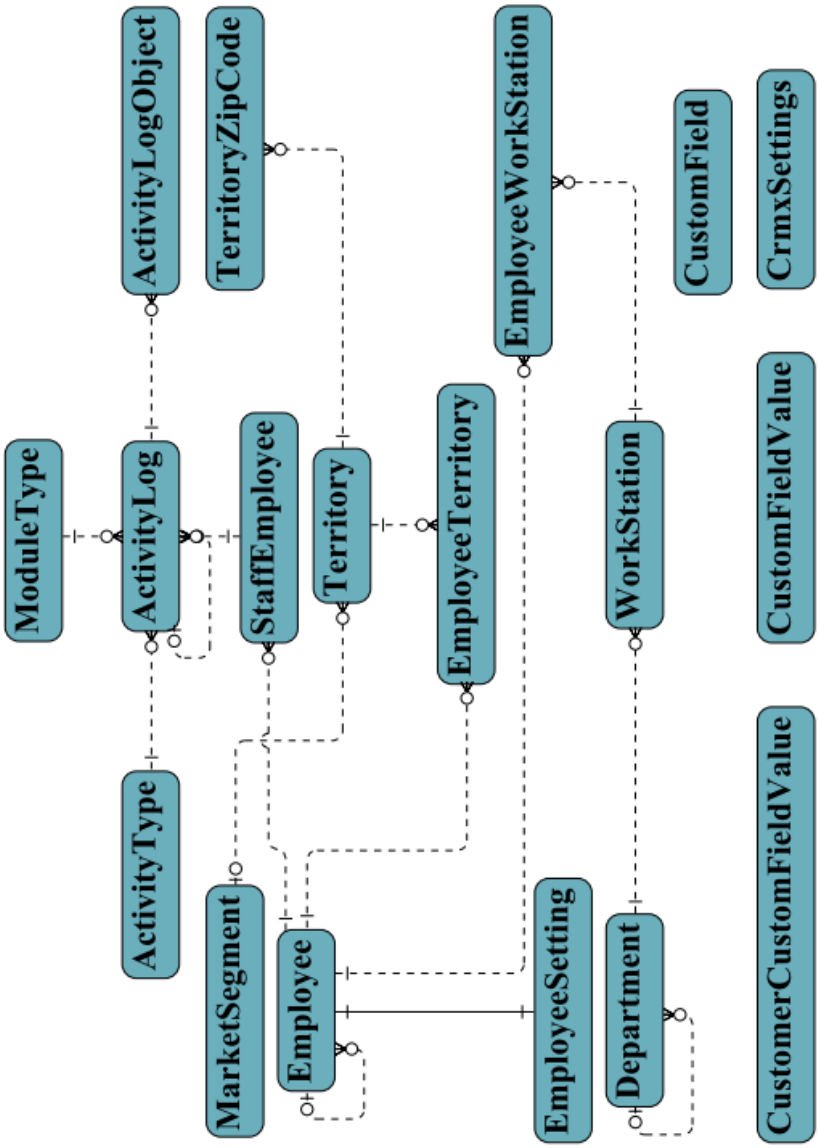

Ilustración 41 Tablas de datos del CRM del Sprint #4

5.2.7.4. Casos de Prueba del Sprint

PB ID: BL031

Nombre: Reportes

Descripción: Ir a Reportes y seleccionar un reporte de tipo “Summary” en “Accounts” luego sin agregar filtros enviar a buscar los datos solicitados.

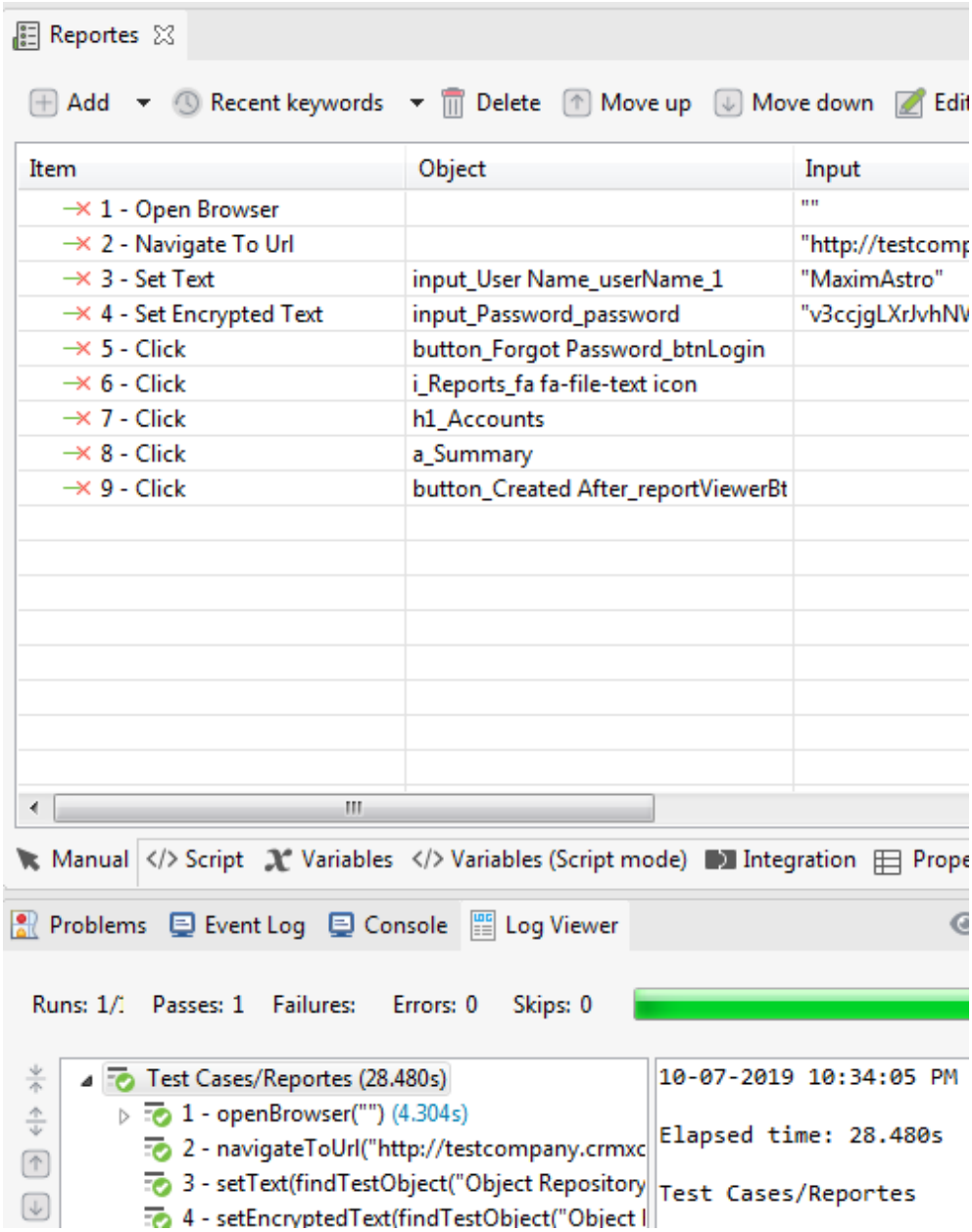

Ilustración 42 Caso de prueba BL031 - Reportes

PB ID: BL011

Nombre: Campos Personalizados

Descripción: Ir a Campos Personalizados agregar un nuevo campo, luego dirigirse a una cuenta cliente y llenar el campo previamente creado y proceder a guardar los cambios.

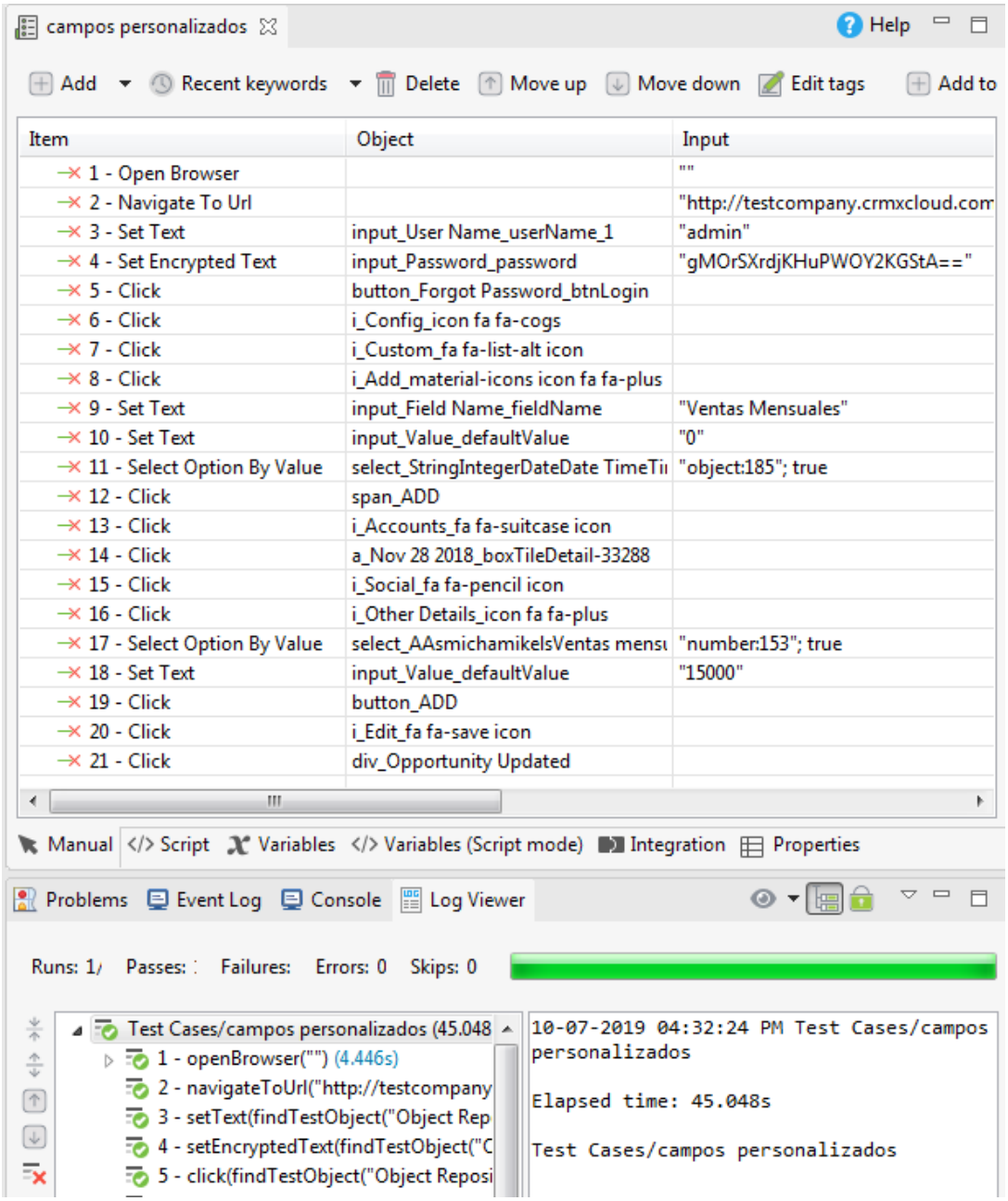

Ilustración 43 Caso de prueba BL011 - Campos Personalizados

PB ID: BL022

Nombre: Panel de Administración

Descripción: Ir a Settings y cambiar valores, luego proceder a almacenar los nuevos cambios en el sistema.

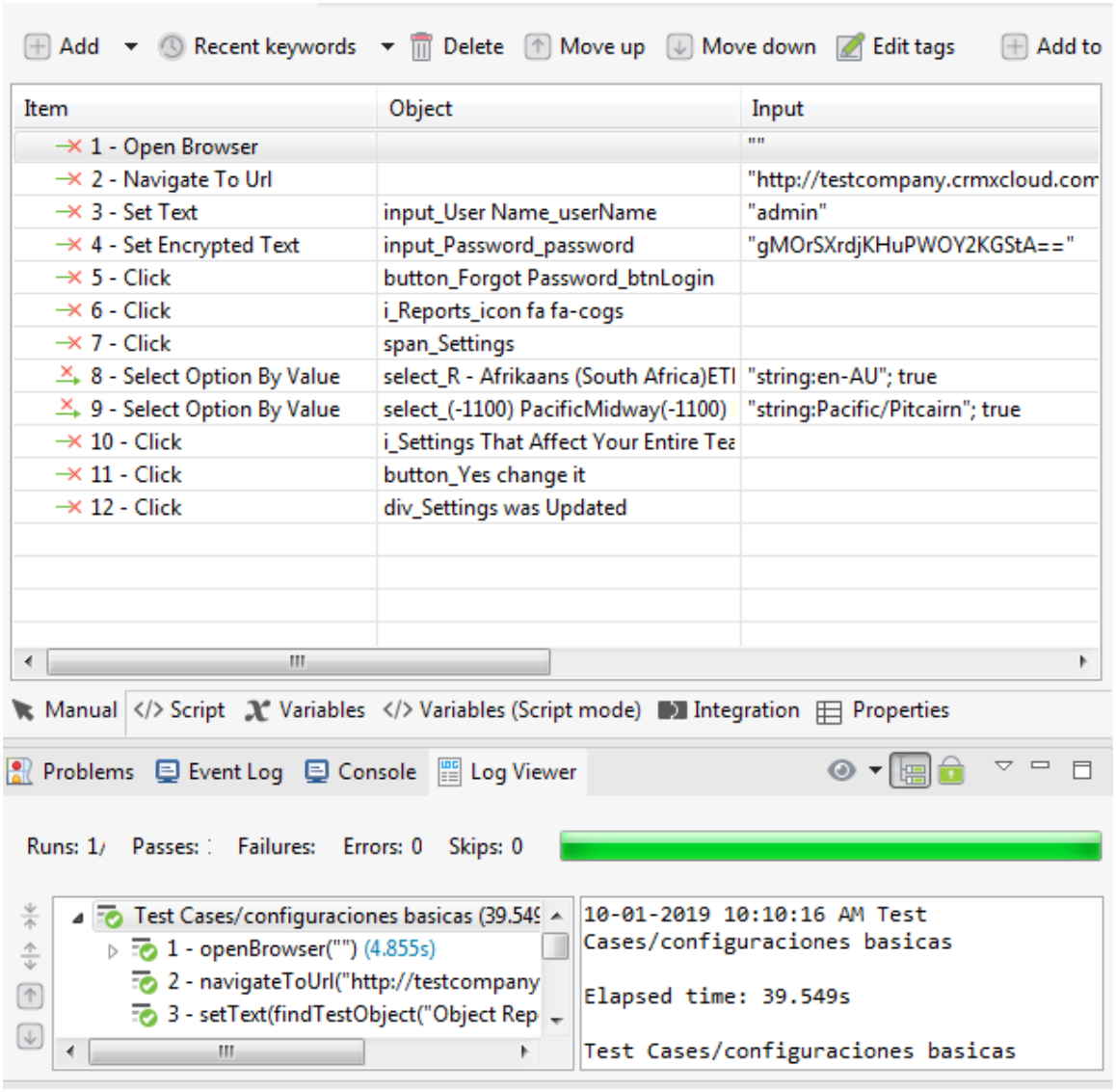

Ilustración 44 Caso de Prueba BL022 - Panel de Administración

PB ID: BL029

Nombre: Roles de Usuario

Descripción: Ir a "*Security Access Mode*l" crear un nuevo rol con permisos reservados.

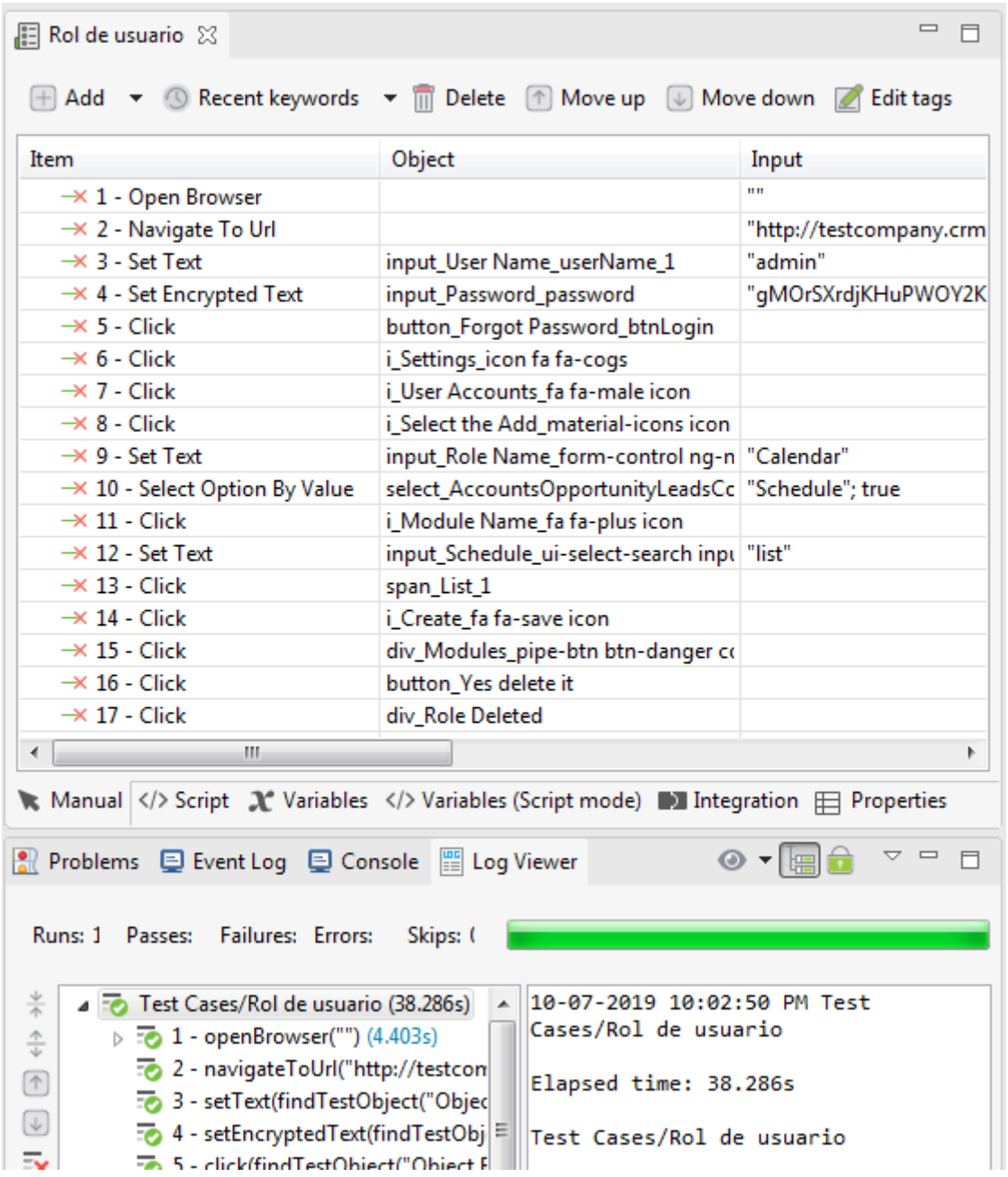

Ilustración 45 Caso de prueba BL029 Roles de Usuario

PB ID: BL033

Nombre: Departamentos y áreas laborales

Descripción: Ir a "*Department*" crear un nuevo departamento, luego crear un puesto y sub departamentos.

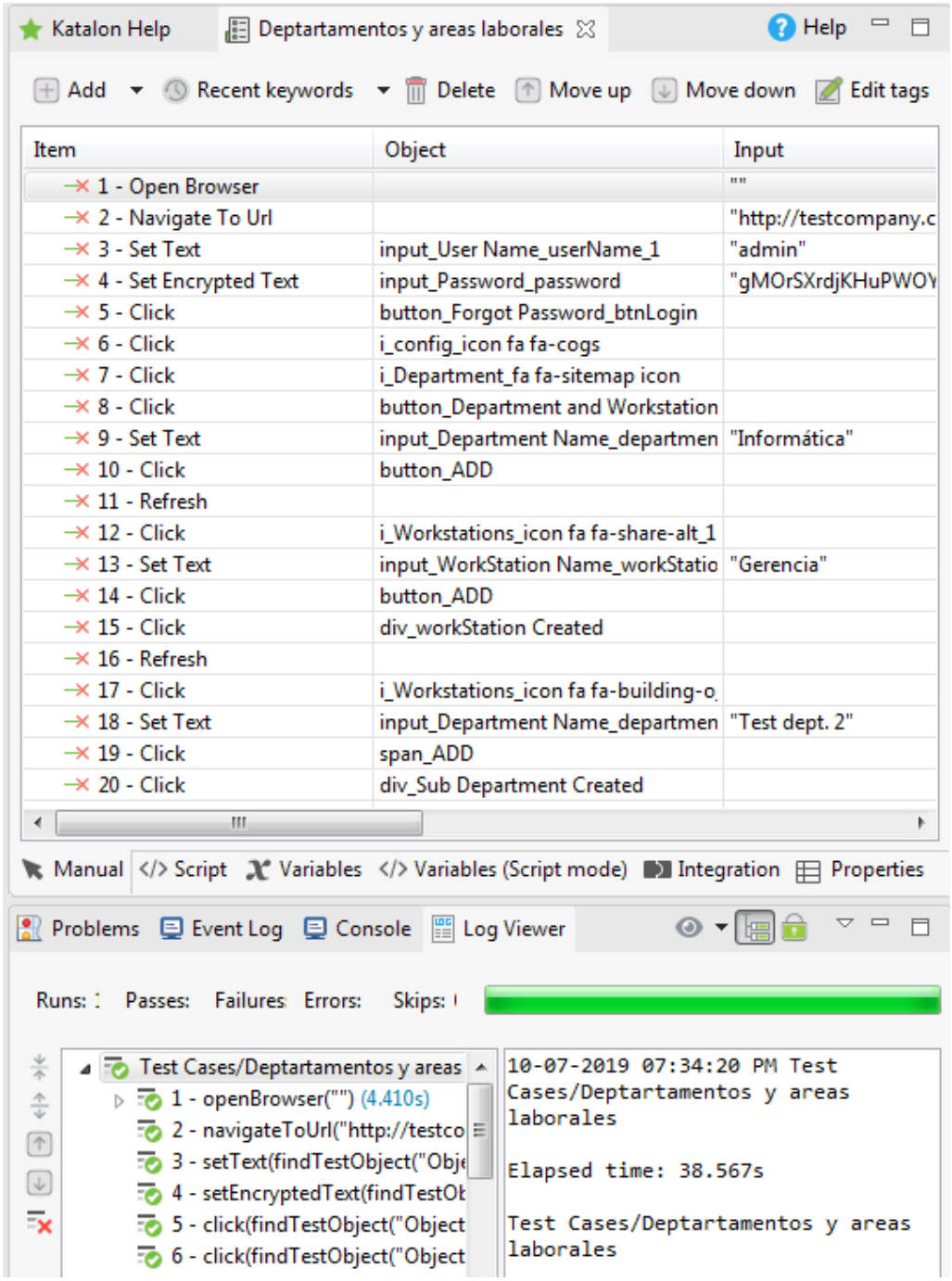

Ilustración 46 Caso de prueba BL033 - Departamentos y áreas laborales

PB ID: BL034

Nombre: Listado de Empleados

Descripción: Ir a "*Employees*" Crear un nuevo empleado y asignarle una estación de trabajo.

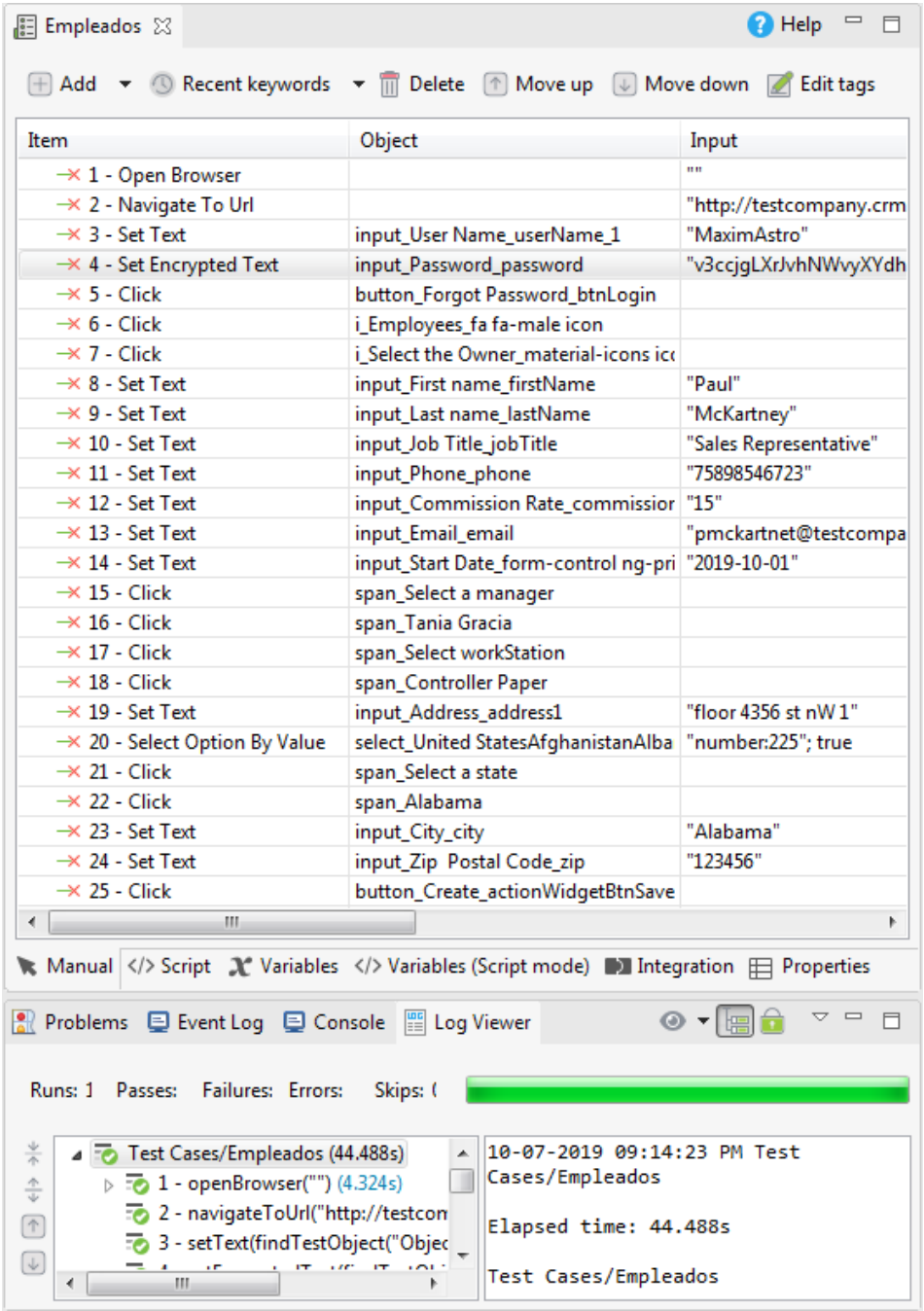

Ilustración 47 Caso de prueba BL034 - Listado de Empleados

5.2.7.5. Revisión del Sprint

Al concluir la semana 19 de trabajo se citó a reunión al equipo Scrum, luego de explicar los avances logrados en el cuarto Sprint y siendo este el ultimo Sprint se procede a terminar toda la documentación perteneciente al sistema para ser utilizada en las capacitaciones y entregadas al Product Owner.

5.2.7.6. Retrospectiva del Sprint

Resultados de retrospectiva del Sprint #4					
Tareas realizadas	8	**Sub-tareas creadas**		18	
Participantes	Equipo Scrum	***Semana***	Inicio semana 16	***Duración***	2 Horas
¿Qué hicimos bien?					
Continuamos la mejora continua en los tiempos de desarrollo lo cual permite resolver problemas de código generados en avances presentados en Sprint anteriores.					
¿Qué mejora puede incorporar el equipo al proceso de forma inmediata?					
-					
¿Qué vamos a hacer para conseguirlo?					
-					

Tabla 14 Retrospectiva de Sprint #4

5.2.8. Product Backlog final

Durante cada Sprint el Product Backlog se fue alimentando de nuevas sub-tareas generadas a partir de cada tarea que debía de ser desarrollada; en la Tabla 15 Product Backlog final se muestran las tareas originales y las sub-tareas identificadas por el prefijo BLS.

ID	Nombre	Estado	Sprint
BL001	Manejo de usuarios	Hecho	1
BL003	Agregar cuentas clientes	Hecho	1

ID	Nombre	Estado	Sprint
BL004	Listar oportunidades por cuentas	Hecho	1
BL005	Detalles de cuenta por oportunidad	Hecho	1
BL006	Crear contactos	Hecho	1
BL007	Filtrar clientes	Hecho	1
BL009	Crear oportunidades	Hecho	1
BLS001	Diseñar y desarrollar pantallas de cuentas	Hecho	1
BLS002	Crear y enlazar tablas de base de datos de cuentas	Hecho	1
BLS003	Preparar y ejecutar caso de prueba	Hecho	1
BLS004	Diseñar y desarrollar pantallas de oportunidades	Hecho	1
BLS005	Crear y enlazar tablas de base de datos de oportunidades	Hecho	1
BLS006	Preparar y ejecutar caso de prueba	Hecho	1
BLS007	Diseñar y desarrollar pantalla de login	Hecho	1
BLS008	Crear y enlazar tabla de usuarios	Hecho	1
BLS009	Crear algoritmos de validación y tokens de seguridad	Hecho	1
BLS010	Preparar y ejecutar caso de prueba	Hecho	1
BLS011	Diseñar y desarrollar pantallas de Contactos	Hecho	1
BLS012	Crear y enlazar tablas de base de datos de contactos	Hecho	1
BLS013	Preparar y ejecutar caso de prueba	Hecho	1
BLS014	Agregar diseño de filtros a pantalla de cuentas	Hecho	1
BLS015	Crear algoritmos de búsqueda por parámetros en cuentas	Hecho	1
BLS016	Mostrar resultados de búsqueda en pantalla de cuentas	Hecho	1
BLS021	Crear pantalla detalles de cuentas	Hecho	1
BLS022	Crear campo en tabla oportunidades de relación a cuentas	Hecho	1
BLS023	En detalle de cuenta mostrar enlaces a	Hecho	1

ID	Nombre	Estado	Sprint
	oportunidades enlazadas		
BLS024	Diseñar y desarrollar pantalla de detalles de oportunidad	Hecho	1
BLS025	En detalle de oportunidad mostrar información de contacto de la cuenta enlazada	Hecho	1
BL010	Multi empresas	Hecho	2
BL012	Notas de clientes	Hecho	2
BL013	Manejar archivos	Hecho	2
BL014	Tareas en cuentas de clientes	Hecho	2
BL015	Detalle de cuenta cliente	Hecho	2
BL016	Cambiar propietario de cuenta cliente	Hecho	2
BL018	Cambio de etapa de oportunidad	Hecho	2
BLS026	Algoritmo de alta de Sub-dominio y DB individual por empresa automatizada	Hecho	2
BLS027	Reenvió de dominio y validación según empresa	Hecho	2
BLS028	Panel de creación de nueva empresa	Hecho	2
BLS029	API de uso y validación de usuario por empresa	Hecho	2
BLS033	Crear pantallas de archivos	Hecho	2
BLS034	Crear tabla archivos y enlazar a controladores	Hecho	2
BLS035	Codificación de crear, listar, eliminar y enlazar archivos con cuentas clientes.	Hecho	2
BLS036	Diseñar y desarrollar pantalla de tareas	Hecho	2
BLS037	Crear tabla de datos para tareas y enlazar a controladores	Hecho	2
BLS038	Codificar algoritmos para crear, enlazar a clientes, editar, eliminar y listar tareas	Hecho	2
BLS039	Diseñar y desarrollar pantalla de detalles	Hecho	2
BLS040	Enlazar a la pantalla todos los objetos relacionados al cliente (contactos, tareas, archivos)	Hecho	2

ID	Nombre	Estado	Sprint
BLS049	Agregar campos de notas a las pantallas detalles cuentas clientes	Hecho	2
BLS050	Crear tabla de notas en la base de datos	Hecho	2
BLS051	Codificar algoritmos para agregar, editar y eliminar notas	Hecho	2
BLS052	Agregar campo de propietario a pantalla edición de cliente	Hecho	2
BLS053	Agregar campo Owner a tablas de clientes en la base de datos	Hecho	2
BLS054	Agregar código de cambio de propietario al editar cliente.	Hecho	2
BLS055	Agregar campo de cambio etapa oportunidad a pantalla detalle oportunidad	Hecho	2
BLS056	Agregar tabla etapas de oportunidad a base de datos y enlazarla con tabla oportunidades	Hecho	2
BLS057	Crear y codificar pantalla para agregar, eliminar, editar tipos de etapas	Hecho	2
BLS058	Agregar código de cambio de etapas	Hecho	2
BL017	Cuentas Leads	Hecho	3
BL019	Actualizar tipo de cuenta Lead/Oportunidad/Cuenta cerrada	Hecho	3
BL021	Oportunidad ganada/perdida	Hecho	3
BL024	Calendario de citas	Hecho	3
BL026	Listar contactos en cuentas cliente	Hecho	3
BL027	Listado de tarea en cuentas cliente	Hecho	3
BL032	Territorios	Hecho	3
BLS059	Diseñar y codificar pantallas Leads	Hecho	3
BLS060	Crear tablas y relaciones en la Base de Datos para Leads	Hecho	3
BLS061	Codificación para crear, editar, mostrar, detallar y eliminar Leads	Hecho	3
BLS062	Agregar columna a tabla oportunidades	Hecho	3

ID	Nombre	Estado	Sprint
BLS063	Codificar controlador para almacenar y actualizar estado	Hecho	3
BLS064	Agregar opción al diseño web	Hecho	3
BLS065	Filtrar contactos por cuentas	Hecho	3
BLS066	Mostrar lista de contactos en pantalla cuentas	Hecho	3
BLS067	Filtrar tareas por cuentas	Hecho	3
BLS068	Mostrar lista de tareas en pantalla cuentas	Hecho	3
BLS069	Crear tablas territorios	Hecho	3
BLS070	Diseñar y desarrollar plantillas de territorios	Hecho	3
BLS071	Agregar botón de cambio de tipo de cuenta	Hecho	3
BLS072	Codificar algoritmo de cambio de estado de la cuenta cliente	Hecho	3
BLS077	Crear tabla de datos para calendario	Hecho	3
BLS078	Crear y desarrollar vista de calendario	Hecho	3
BLS079	Agregar, editar, eliminar, actualizar actividades de calendario	Hecho	3
BLS080	Agregar características calendarios a módulos requeridos	Hecho	3
BL011	Campos personalizados	Hecho	4
BL022	Panel de administración	Hecho	4
BL029	Roles de usuarios	Hecho	4
BL031	Reportes	Hecho	4
BL033	Departamentos y áreas laborales	Hecho	4
BL034	Listado de empleados	Hecho	4
BLS081	Diseñar y desarrollar plantillas de reportes	Hecho	4
BLS082	Integrar plantillas al sistema, sección reportes	Hecho	4
BLS083	Codificación de filtros y enlaces a base de datos	Hecho	4
BLS084	Diseñar y desarrollar pantalla de	Hecho	4

ID	Nombre	Estado	Sprint
	campos personalizados		
BLS085	Crear tabla de datos para campos personalizados	Hecho	4
BLS086	Codificación para agregar, editar, eliminar campos personalizados de tipo múltiple	Hecho	4
BLS087	Diseñar y desarrollar pantalla de panel de administración	Hecho	4
BLS088	Crear tablas de datos de configuraciones y enlazar a controlador	Hecho	4
BLS089	Codificar edición de ajustes	Hecho	4
BLS090	Crear tablas departamentos y áreas laborales	Hecho	4
BLS091	Diseñar y desarrollar plantillas de áreas laborales y departamentos	Hecho	4
BLS092	Enlazar a módulo de empleados	Hecho	4
BLS093	Crear tabla en base de datos para empleados	Hecho	4
BLS094	Diseñar y desarrollar plantillas web para empleados	Hecho	4
BLS095	Codificar eliminar, agregar, editar y actualizar empleados.	Hecho	4
BLS096	Crear tablas de roles y permisos	Hecho	4
BLS097	Codificar pantallas para asignar roles y permisos	Hecho	4
BLS098	Codificar algoritmos de distribución de accesos por usuarios según sus permisos	Hecho	4

Tabla 15 Product Backlog final

Ver en Anexos la tabla Web Services que contiene el listado de Servicios implementados en el proceso de desarrollo del sistema.

5.3. Implantación del sistema

Para instalar y poner en marcha el sistema fue necesario hacer uso de dos productos de AWS (Amazon Web Service), una máquina virtual con Windows Server y un servicio de bases de datos para SQL Server.

5.3.1. Servidor Web

La Máquina Virtual que se ha creado es del tipo EC2 (Elastic Computer Cloud), cuenta con el sistema operativo de Windows Server 2012 Estándar y ofrece la capacidad de fácil escalabilidad según crezcan las necesidades de recursos del sistema.

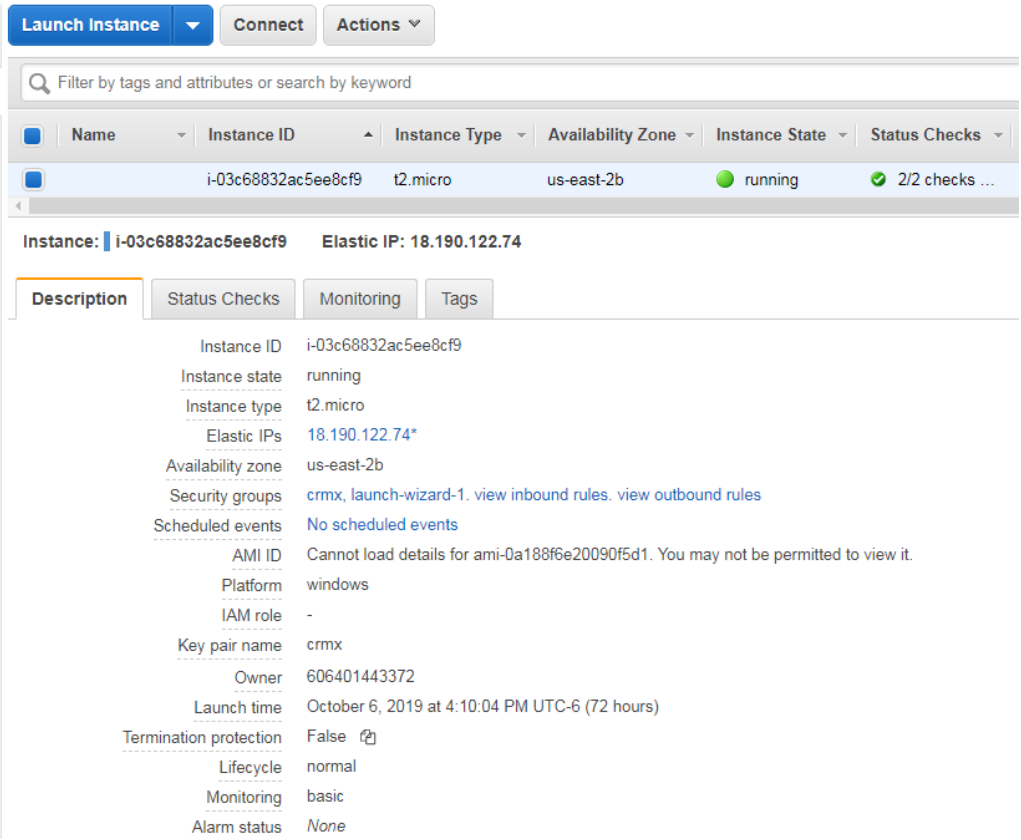

Ilustración 48 Descripción de la instancia de Máquina Virtual corriendo en AWS

El siguiente paso fue importar al servidor los archivos necesarios para el funcionamiento del sistema, cada servicio fue previamente compilado en Visual Studio y luego fue ubicado en el directorio que le corresponde dentro del servidor web.

En la *Ilustración 49 Directorio de Archivos del API* (Application Programming Interface) se muestra la raiz de los recursos que sirven como interfaz para la facil comunicación entre el sistema operativo, base de datos o protocolos de comunicaciones.

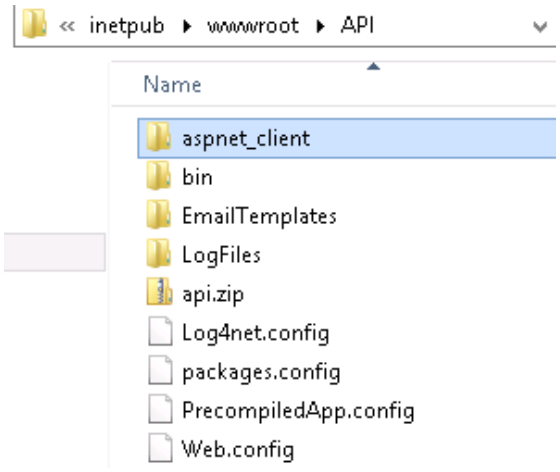

Ilustración 49 Directorio de Archivos del API

La *Ilustración 50 Directorio de Archivos del SPA* (Single Page Application) muestra los recuersos que permiten la operación de la interfaz grafica.

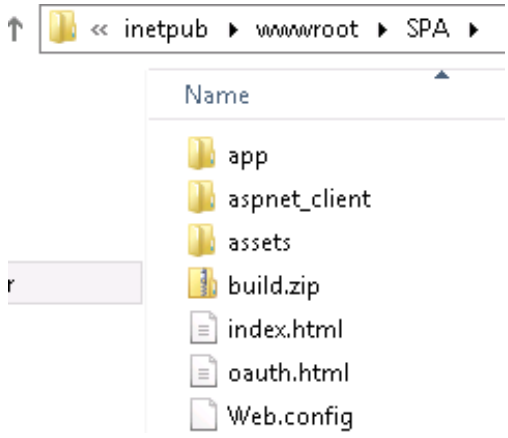

Ilustración 50 Directorio de Archivos del SPA

Ilustración 51 Directorio de Archivos de FSS (File System Service) muestra los recursos a consumir y que permitiran la gestión de los archivos que cada usuario cargue a traves de la interfaz grafica del sistema.

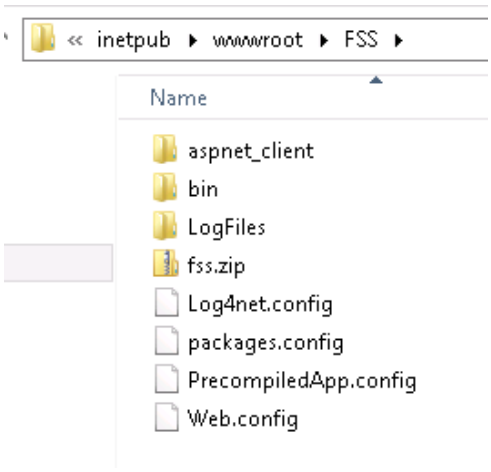

Ilustración 51 Directorio de Archivos de FSS

Por ultimo, en la Ilustración 52 Directorio de Archivos de SMA (Security Model Access) se muestran los recursos destinados a gestionar la seguridad de la plataforma.

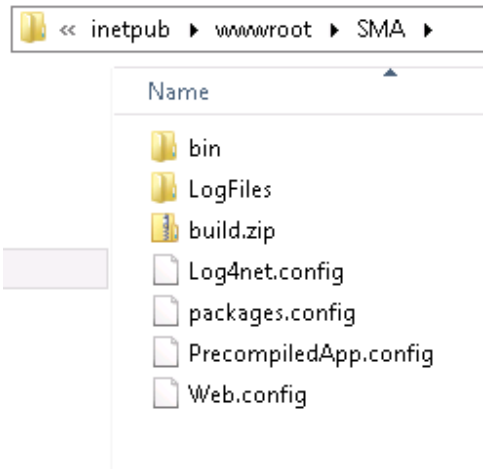

Ilustración 52 Directorio de Archivos de SMA

Con el fin de hacer sencillo para los usuarios el acceso al sistema, en la plataforma de administración del dominio *crmxcloud.com* se agregó la IP publica de acceso que otorga AWS a la máquina virtual, este ajuste se puede observar en la Ilustración 53.

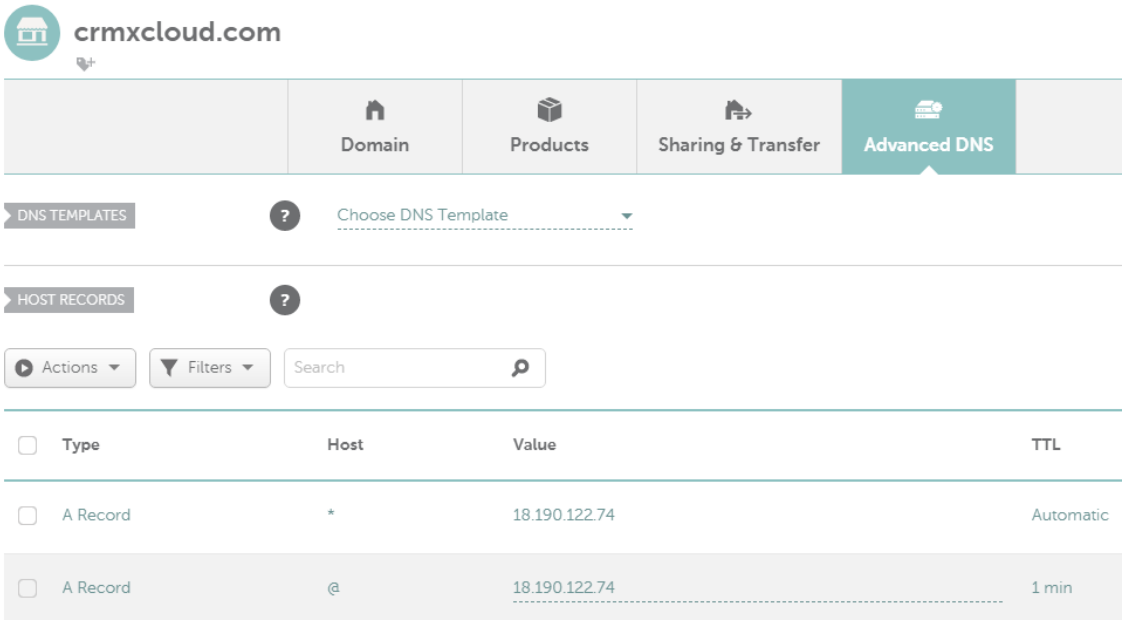

Ilustración 53 Configuración de IP de destino en el dominio

El siguiente paso fue activar y configurar el servicio IIS permitiendo así el correcto funcionamiento de las solicitudes del dominio web hacia el sistema, en la *Ilustración 54 URL de acceso al sistema CRMX* se muestran las conexiónes permitidas y que estan enlazadas a cada uno de los directorios previamente creados (API, FSS, SMA y SPA).

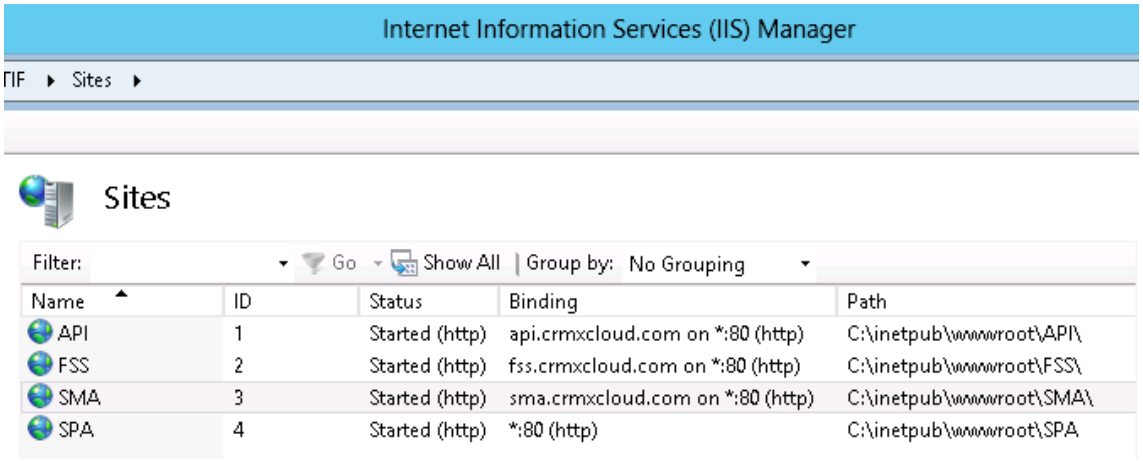

Ilustración 54 URL de acceso al sistema CRMX

5.3.2. Base de Datos

Para la base de datos se ha creado un servicio RDS (Relational Database Service) de AWS que facilita la configuración, la utilización y el escalado de las implementaciones de SQL Server.

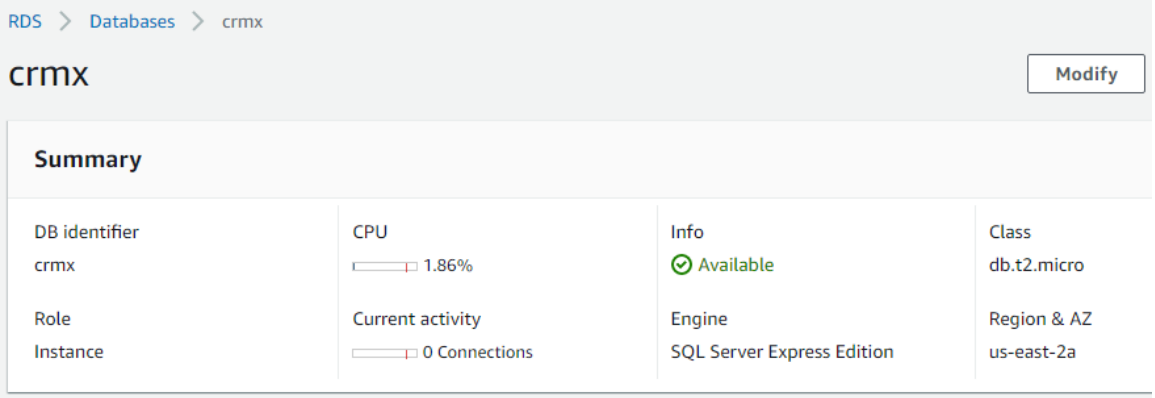

Ilustración 55 Detalles del servicio RDS en AWS

Luego de levantar el servicio RDS procedimos a configurar usuario y contraseña de accesos a SQL Server para permitir al sistema conectarse y realizar todas las consultas pertinentes a la base de datos.

Para crear todas las tablas necesarias para cada Tenant el equipo de desarrollo ha creado un Script de tipo *"Code First"* que al recibir una solicitud de un nuevo Tenant ejecuta la orden de crear dentro de la base de datos cada tabla llamando a otro Script específico para cada tabla como se muestra en la lista de la Ilustración 56; este código se ejecuta en los directorios SMA y API lo que permite la fácil implementación de un nuevo Tenant sin necesidad de intervención de un programador.

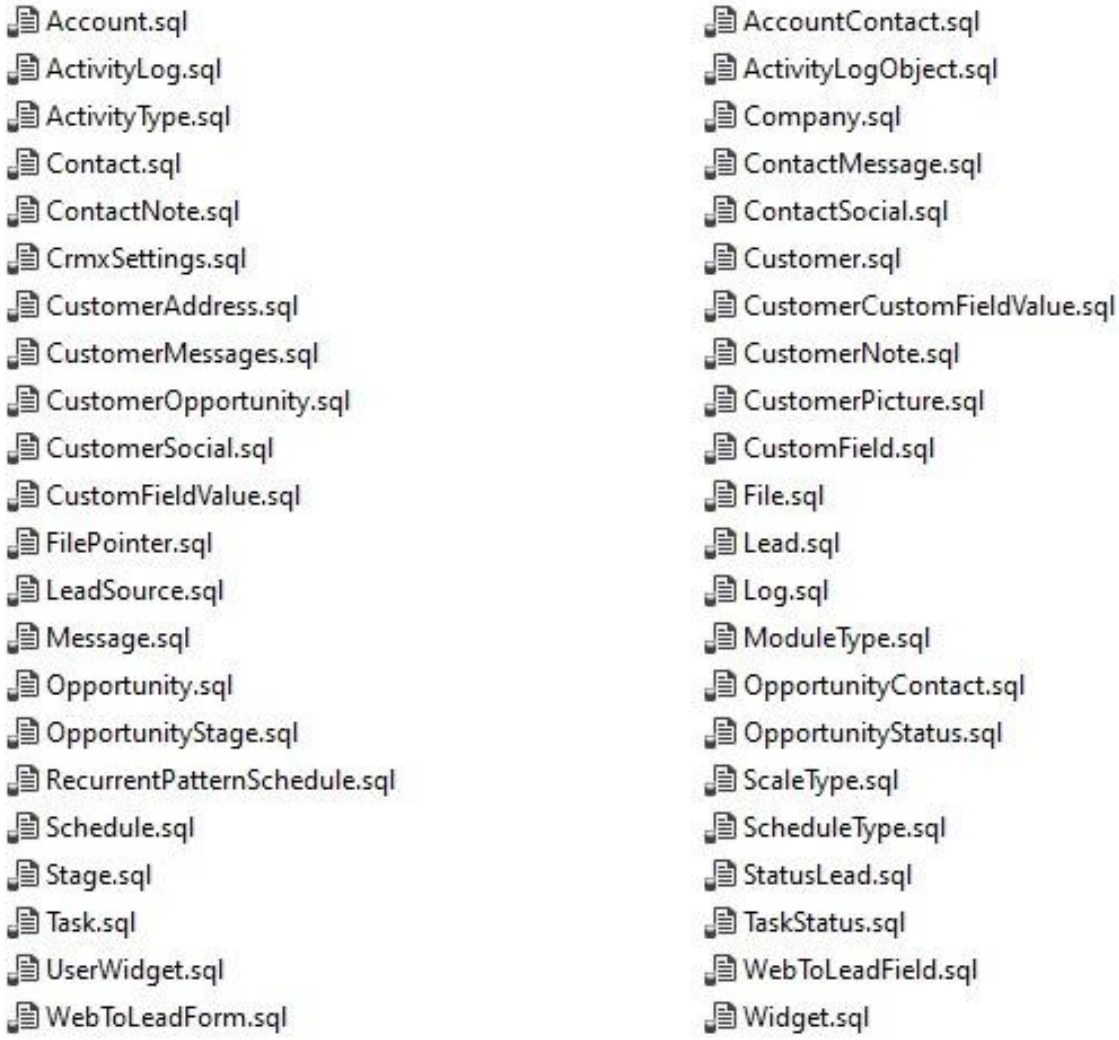

Ilustración 56 Scripts SQL a ejecutar al momento de crear un nuevo Tenant

Para finalizar y lograr que los archivos de sistema y base de datos respondan a la comunicación dominio-servidor se configuró el archivo "*web.config"* en el directorio API para agregar los datos correspondientes al dominio web (crmxcloud.com) y la información de conexión a la base de datos, el siguiente paso fue reiniciar el Manejador de Sitios para operar con la nueva configuración.

En Anexos se encuentra la para poner en marcha el sistema CRMX en nuevos servidores. La guía se encuentra en idioma ingles dado que es orientada para los encargados de la implantación del sistema por requerimiento de la empresa UBS que radica en Estados Unidos de América.

5.4. Capacitación del personal

Tomando en cuenta que el personal que realiza uso del sistema CRMX se encuentra ubicado en Estados Unidos de Norte América, UBS oriento hacer uso de la aplicación de video llamadas Skype para realizar la capacitación, en la cual participo un equipo de colaboradores conformado por cinco (5) personas, una por cada área usuaria del sistema, siendo estas, los departamentos de Cuentas, Oportunidades, Leads, Informática y Gerencia.

Calendario de Capacitación					
Temas	**Días**				
	Lunes	**Martes**	**Miércoles**	**Jueves**	**Viernes**
Personalización	*X*				
Leads	*X*				*X*
Oportunidades		*X*			*X*
Cuentas			*X*		*X*
Tareas			*X*		
Contactos				*X*	
Empleados				*X*	
Calendario					*X*
Nuevo Tenant		*X*			
Usuarios y roles	*X*				
Archivos			*X*		
Reportes			*X*		
Departamentos y áreas laborales				*X*	

Tabla 16 Calendario de Capacitación

La capacitación tuvo una duración de cinco días dado el conocimiento previo en sistemas CRM que ya poseían los usuarios asignados a la capacitación. En cada día se destinaron tres horas laborales de capacitación.

6. Conclusiones

Luego de haber aplicado Scrum como metodología de desarrollo junto con las herramientas de desarrollo descritas en el Marco Teórico, podemos concluir lo siguiente:

- Se cumplieron todos y cada uno de los objetivos establecidos para el trabajo monográfico, desde la identificación de requerimientos hasta la implantación del sistema web en los servidores Amazon Web Services, quedando el sistema en producción en la empresa Ultimate Business Services. El cierre de las oficinas de la empresa PartsBase, Inc. Nicaragua en el 2018 representó una fuerte limitante para hacer de nuestro conocimiento si el sistema CRMX está siendo utilizado por las demás empresas del grupo R Hammond L.P., lo que no invalida el total cumplimiento de los objetivos del trabajo monográfico.
- Uno de los grandes logros y satisfacciones personales al desarrollar el trabajo monográfico fue el haber entendido Scrum con la suficiente claridad como para poder aplicarlo sistemáticamente de forma competente, logrando un proceso de mejora continua y de innovación en el sistema desarrollado.
- El Product Backlog inicial del proyecto y la información recopilada a partir de la exploración a los sistemas anteriormente en uso, fueron los principales insumos para la Identificación de los requerimientos del sistema, la planeación y desarrollo de los sprint, y demás tareas del ciclo de vida de Scrum.
- El uso de las tecnologías de desarrollo ASP.Net, C# y Angular fueron de gran ayuda para lograr la escritura de código limpio, organizado y modular, lo que hace que sea más fácil para el sistema CRMX escalar en términos de funcionalidad.
- Las pruebas realizadas con Katalon Studio permitieron garantizar el correcto funcionamiento del sistema, a lo largo del desarrollo de cada uno de los sprint.

- La capacitación del sistema se realizó en el idioma inglés que es la lengua nativa de los usuarios, utilizando Skype como herramienta de comunicación interactiva entre usuarios y capacitadores.
- Trabajar con Amazon Web Services facilita el acceso al sistema web CRMX desde diversos dispositivos y navegadores modernos, en cualquier momento y lugar, dada su alta disponibilidad y escalabilidad.

Haber concluido el desarrollo e implantación del sistema web CRMX ha sido una experiencia enriquecedora en el afianzamiento de conocimientos y habilidades adquiridas durante los años de estudio en la Universidad Nacional de Ingeniería.

7. Recomendaciones

Con el fin de mejorar la operabilidad y eficiencia del sistema CRMX, recomendamos las siguientes acciones:

- El desarrollo de un módulo para la gestión de administración de los Tenant, a fin de facilitar la creación, actualización y eliminación de los distintos Tenant del CRMX.
- Desarrollar un módulo para auditar el uso del sistema en operación. Este módulo podrá consumir los registros de la tabla *Activity Log* ubicada en la base de datos CRM.
- Implementar un plan de contingencia ante vulnerabilidades y fallos que pueda presentar Windows Server 2012 que son ajenos al sistema implantado.
- Incorporar certificados SSL para ofrecer protección a los datos transferidos durante el uso del sistema. El uso de certificados SSL permite generar credibilidad y confianza en el sitio.

8. Bibliografía

Acerca de HTML. (s.f.). Obtenido de http://www.acercadehtml.com/manual-html/que-es-html.html

Amazon AWS. (2019). Obtenido de RDS: https://aws.amazon.com/es/rds/

AWS Amazon. (2019). Obtenido de https://aws.amazon.com/es/ec2/

Biblioteca Profesional. (10 de Enero de 2016). Obtenido de http://bibliotecaprofesional.com/que-es-sql-server/

Bustamante Granda, W. X. (2017). Multitenancy. *INNOVA Research Journal*, págs. 72-83.

Crozdesk TM. (s.f.). Obtenido de Crozdesk TM: https://crozdesk.com/it/mockups-prototyping-software/mockflow

EcuRed. (20 de septiembre de 2010). Obtenido de https://www.ecured.cu/Microsoft_Visual_Studio

Every Interaction. (s.f.). Obtenido de Wireframes: https://www.everyinteraction.com/definition/wireframes/

Forma Talet - Bussines School. (s.f.). Obtenido de http://formatalent.com/que-es-angular-en-programacion/

Francia, J. (17 de agosto de 2017). *SCRUM.ORG TM*. Obtenido de SCRUM.ORG TM: https://www.scrum.org/resources/blog/scrum-no-es-una-metodologia-es-un-framework

Goikolea, M. (19 de Mayo de 2014). *iebschool*. Obtenido de https://www.iebschool.com/blog/que-es-saas-definicion-ventajas-digital-business/

Goldszmidt, G. (11 de 10 de 2011). *IBM Developer*. Obtenido de https://www.ibm.com/developerworks/ssa/webservices/library/ws-middleware/index.html

Gonzales, M. (19 de Diciembre de 2017). *DIMAJEFF*. Obtenido de DIMAJEFF: http://www.dimajeff.com.mx/blog/articulos/item/41-que-es-scrum

Gutiérrez Medina, G. (25 de Julio de 2018). *PíldorasUX*. Obtenido de https://medium.com/pildorasux/qu%C3%A9-es-un-wireframe-a5f49d8c81d

Gutiérrez, J. J. (2005). *Lenguajes y sistemas informaticos.* Obtenido de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

GxTechnical. (s.f.). *GxTechnical*. Obtenido de Developers Tools: http://library.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web_Services.htm

Microsoft Docs. (19 de 07 de 2015). Obtenido de https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework

Morales, M. (29 de Enero de 2016). *Javier Garzas*. Obtenido de http://www.javiergarzas.com/2016/01/serie-cloud-for-dummies-aws-que-es-eso.html

Muñoz Serafín, M. (1 de Diciembre de 2015). *Miguel Muñoz Serafín*. Obtenido de https://msmdotnet.wordpress.com/2015/12/01/implementando-web-api-en-asp-net-mvc/

Portal SaaS. (s.f.). Obtenido de http://www.portalsaas.com/que-es-saas-software-as-a-service.php

ProAgilist. (22 de abril de 2016). *ProAgilist*. Obtenido de https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/los-11-pasos-para-implementar-metodologia-scrum/

Salesforce. (10 de Enero de 2018). Obtenido de from https://www.salesforce.com/crm/what-is-crm/

Schwaber, K. S. (2012). *Software In 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust.* USA: Jhon Wiley & Sons, Inc.

Schwaber, K., & Sutherland, J. (2012). *Software In 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust.* USA: Jhon Wiley & Sons, Inc.

Skype. (s.f.). Obtenido de https://support.skype.com/es/faq/FA6/que-es-skype

Souvenirs de IT. (s.f.). Obtenido de ASP.NET Web API, paso a paso - 1 - ¿Qué es Web API?: https://itsouvenirs.blogspot.com/2016/03/aspnet-web-api-paso-a-paso-1.html

Straud, B., & Chaco, S. (2014). *Pro Git.* California, USA: Apress.

Systems, O. (12 de Julio de 2015). *Orix.* Obtenido de https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza

Yang, J. (2017). *Utest*. Obtenido de https://www.utest.com/tools/katalon

Anexos

Costos del proyecto

Para la estimación de los costos del proyecto se tomaron como referencia los costos de uso del software y servidores, publicados en los respectivos sitios oficiales.

Por su parte, la precisión de los costos se eleva a la decena más alta, con el fin de no manejar unidades ni decimales.

Software

Nombre	Cantidad	Meses	Precio por mes	Total $ USD
Visual Studio	2	5	45.00	450.00
Bitbucket	2	5	3.00	30.00
Mockflow	2	5	14.00	140.00
Katalon Studio	2	5	127.00	1,270.00
ASP. Net Web Api	2	5	0.00	0.00
C#	2	5	0.00	0.00
Angular	2	5	0.00	0.00
Bootstrap	2	5	0.00	0.00
HTML	2	5	0.00	0.00
			Total	**1,890.00**

Tabla 17 Costos de licencias de Software

Servidores Amazon Web Services

Nombre	Cantidad	Meses	Precio por mes	Total $ USD
EC2: Windows Server 2012	1	5	90.00	450.00
RDS: SQL Server	1	5	100.00	500.00
			Total	**950.00**

Tabla 18 Costo de uso de Amazon Web Services

Recursos Humanos

Dado que el Sistema Web CRMX no era el único desarrollo a cargo del jefe de proyecto, se prorratea su salario entre 4 proyectos que se ejecutaban simultáneamente en la empresa PartsBase Inc.

Cargo	Proyectos	Salario base $	Salario por proyecto $ USD
Jefe de proyectos	4	1,300	325.00

Tabla 19 Salario del jefe de proyectos

Nombre	Cantidad	Meses	Salario mensual	Total $ USD
Programadores	2	5	750.00	7,500.00
Jefe de proyecto	1	5	325.00	1,625.00
			Total	**9,125.00**

Tabla 20 Costo de Recursos Humanos

Costos complementarios

Nombre	Cantidad	Meses	Precio	Total $ USD
Dominio Web	1	12	18.00	18.00
			Total	**18.00**

Tabla 21 Costos complementarios

Costo total

Nombre	Total $ USD
Licencias de Software	1,890.00
Uso de AWS	950.00
Recursos Humanos	9,125.00
Costos complementarios	18.00
Total	**$11,983.00**

Tabla 22 Costo total del proyecto

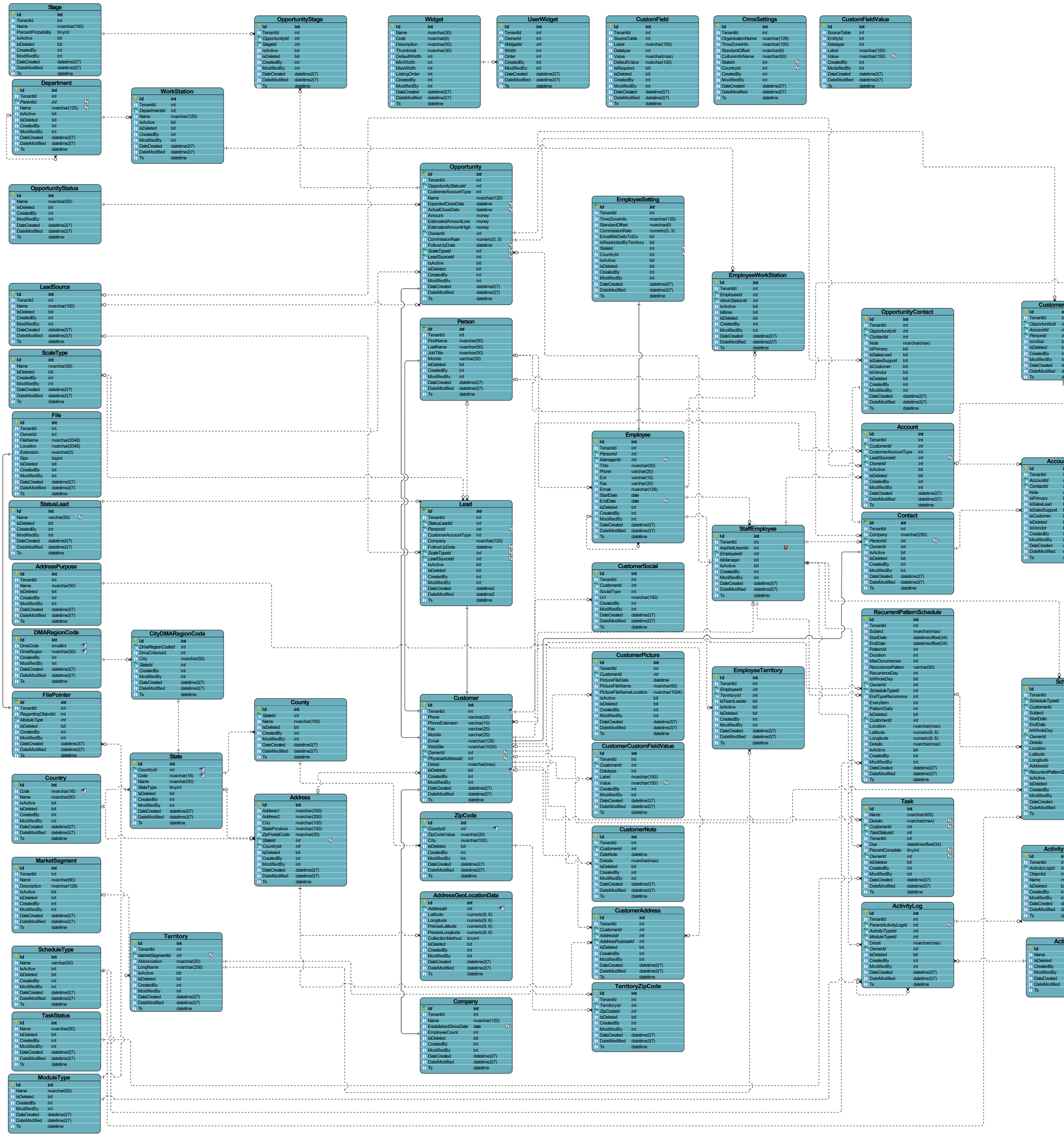

Ilustración 57 Modelo Físico Base de Datos CRM

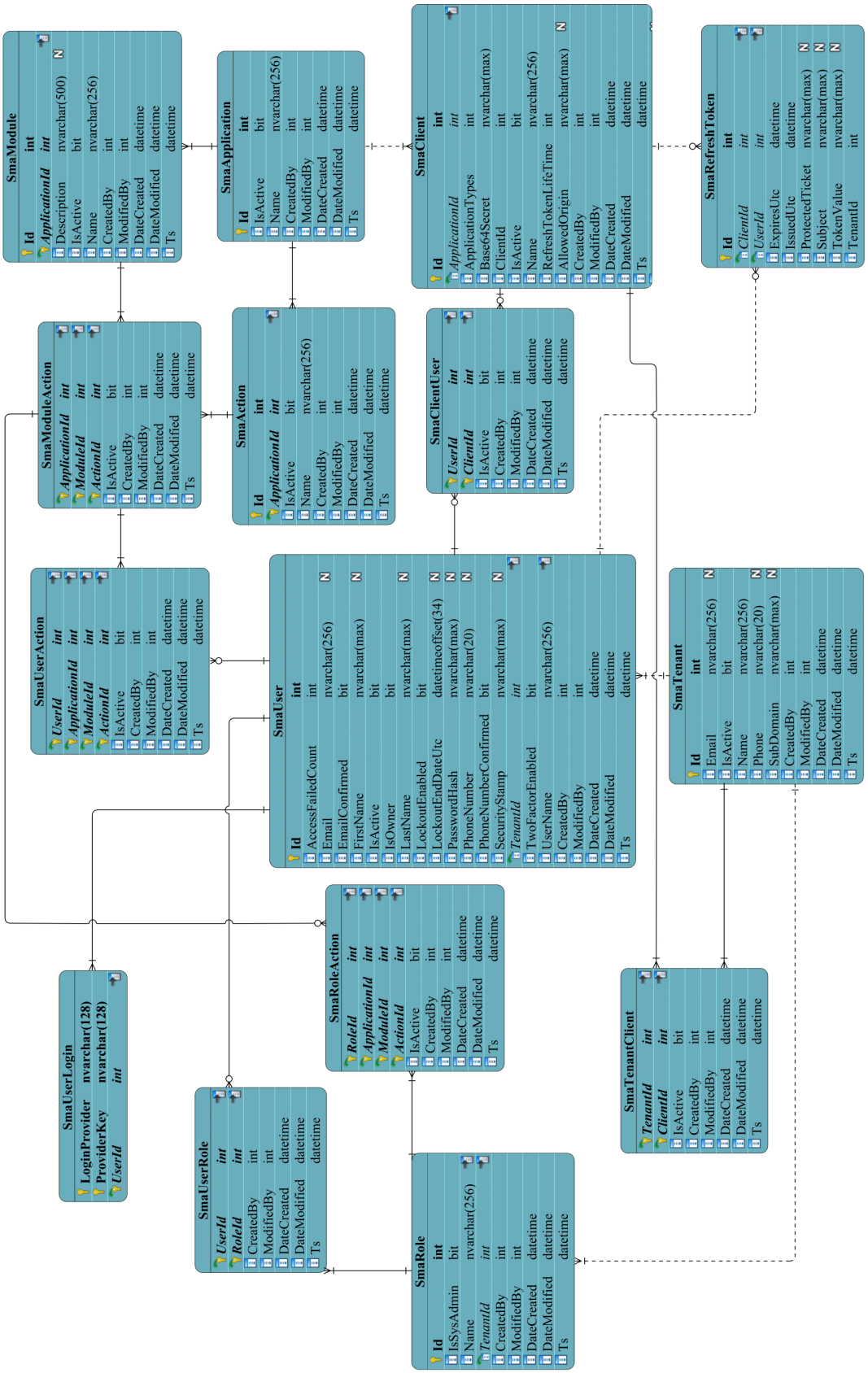

Ilustración 58 Modelo Físico de la Base de Datos SMA

Diccionario de Datos

Sprint #1

Base de datos del SMA

Nombre de la tabla	**Descripción de la tabla**	
SmaClientUser	**tabla utilizada para asociar un usuario con un cliente del sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
IsActive	Permite identificar si el registro está activo	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
UserId	FK (SmaUser)	Int
ClientId	FK (SmaClient)	Int
CreatedBy	FK (SmaUser)	Int
ModifiedBy	FK (SmaUser)	Int

Tabla 23 Diccionario de datos tabla SmaClientUser

Nombre de la tabla	**Descripción de la tabla**	
SmaRefreshToken	**tabla utilizada para los token de extensión de sesión**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del RefreshToken	Int
ExpiresUtc	Fecha de expiración de la sesión del usuario	datetime
IssuedUtc	Fecha de inicio de sesión del usuario	datetime
ProtectedTicket	Hash que contiene información de la nueva sesión usuario	nvarchar
Subject	Nombre de usuario que solicita la extensión de sesión	nvarchar
TokenValue	Valor del token para extender la sesión	nvarchar
UserId	FK (SmaUser)	Int
ClientId	FK (SmaClient)	Int
TenantId	FK (SmaTenant)	Int

Tabla 24 Diccionario de datos tabla SmaRefreshToken

Nombre de la tabla	**Descripción de la tabla**	
SmaUser	**tabla utilizada para los usuarios del sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del usuario	Int
AccessFailedCount	Numero de intentos fallidos para inicio de sesión	Int
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
Email	Correo electrónico del usuario	nvarchar
EmailConfirmed	Permite identificar si el email fue confirmado	Boolean
FirstName	Almacena el primer nombre del usuario	nvarchar
IsActive	Permite identificar si el registro está activo	Boolean
IsOwner	Utilizado para identificar si el usuario es propietario del tenant	Boolean
LastName	Almacena el apellido del usuario	nvarchar
LockoutEnabled	Identifica si el usuario fue bloqueado	Boolean
LockoutEndDateUtc	Fecha de bloqueo del usuario	datetimeoffset
PasswordHash	Contraseña de usuario encriptada	nvarchar
PhoneNumber	Número de teléfono del usuario	nvarchar
PhoneNumberConfirmed	Teléfono fue confirmado	Boolean
SecurityStamp	Hash de la última sesión del usuario	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
UserName	Nombre de usuario para el inicio de sesión	nvarchar
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 25 Diccionario de datos tabla SmaUser

Base de datos del CRM

Nombre de la tabla	**Descripción de la tabla**	
Account	**Utilizada para gestionar las cuentas de la compañía**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de las cuentas de clientes de la empresa	Int
DateCreated	Almacena la fecha de creación	datetime2

Nombre de la tabla	**Descripción de la tabla**	
Account	**Utilizada para gestionar las cuentas de la compañía**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Permite identificar si la cuenta esta activa	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
CustomerAccountType	Identifica la cuenta como comercial o residencial	Int
Ts	Almacena información relacionada a la zona horaria	datetime
CreatedBy	**FK (SmaUser)**	Int
CustomerId	**FK (Customer)**	Int
LeadSourceId	**FK (LeadSource)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OwnerId	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 26 Diccionario de datos tabla Account

Nombre de la tabla	**Descripción de la tabla**	
AccountContact	**tabla utilizada para la relación entre las tablas Account y Contact**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del AccountContact	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsCustomer	Marca el contacto como un Customer	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
IsPrimary	Define si el contacto es el principal	Boolean
IsSalesLead	Marca el contacto como Sales lead	Boolean
IsSalesSupport	Define si el contacto es de soporte a ventas	Boolean
IsVendor	Define si el contacto es el vendedor	Boolean
Note	Permite agregar una nota a la creación del contacto	nvarchar
Ts	Almacena información relacionada a la zona horaria	datetime
AccountId	**FK (Account)**	Int
ContactId	**FK (Contact)**	Int

Nombre de la tabla	**Descripción de la tabla**	
AccountContact	**tabla utilizada para la relación entre las tablas Account y Contact**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 27 Diccionario de datos tabla AccountContact

Nombre de la tabla	**Descripción de la tabla**	
Company	**Contiene datos acerca de la compañía**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la compañía	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
EmployeeCount	Almacena el contador del número de empleados	Int
EstablishedSinceDate	Fecha de fundación de la compañía	date
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Almacena el nombre de la compañía	nvarchar
Ts	Almacena información relacionada a la zona horaria	datetime
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 28 Diccionario de datos tabla Company

Nombre de la tabla	**Descripción de la tabla**	
Contact	**tabla relacionada para la creación de contactos**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Almacena el identificador del registro	Int
Company	Nombre de la empresa a la que pertenece el contacto	nvarchar
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Permite identificar si el contacto está activo	Boolean

Nombre de la tabla	**Descripción de la tabla**	
Contact	**tabla relacionada para la creación de contactos**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Almacena información relacionada a la zona horaria	datetime
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OwnerId	**FK (SmaUser)**	Int
PersonId	**FK (Person)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 29 Diccionario de datos tabla Contact

Nombre de la tabla	**Descripción de la tabla**	
Customer	**Almacena datos relacionado a los clientes**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Customer	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Detail	Permite agregar detalles relacionados al cliente	nvarchar
Email	Almacena el correo electrónico de la persona	nvarchar
Fax	Almacena el número de fax de la persona	varchar
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Mobile	Almacena el número móvil de la persona	varchar
Phone	Almacena el número telefónico de la persona	varchar
PhoneExtension	Almacena el número de extensión telefónico de la persona	varchar
Ts	Almacena información relacionada a la zona horaria	datetime
WebSite	Sitio web del Customer	nvarchar
PhysicalAddressId	**FK (Address)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OwnerId	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 30 Diccionario de datos tabla Customer

Nombre de la tabla	**Descripción de la tabla**	
CustomerOpportunity	**Tabla utilizada para la relación entre las tablas Customer y Opportunity**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la relación CustomerOpportunity	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
IsInitial	Marca al registro CustomerOpportunity como inicial	Boolean
Ts	Almacena información relacionada a la zona horaria	datetime
AccountId	**FK (Account)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OpportunityId	**FK (Opportunity)**	Int
PersonId	**FK (Person)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 31 Diccionario de datos tabla CustomerOpportunity

Nombre de la tabla	**Descripción de la tabla**	
Opportunity	**Tabla utilizada para almacenar los datos de las Oportunidades**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la Oportunidad	Int
ActualCloseDate	Fecha de cierre estimada	datetime
Amount	Costo de la oportunidad	money
CommissionRate	Porcentaje de comisión	numeric
CreatedBy	Almacena el Id del usuario que ingreso el registro	Int
CustomerAccountType	Identifica la oportunidad como comercial o residencial	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
EstimatedAmountHigh	Costo máximo estimado	money
EstimatedAmountLow	Costo mínimo estimado	money
ExpectedCloseDate	Fecha de cierre estimada	datetime
FollowUpDate	Fecha de seguimiento	datetime

Nombre de la tabla	**Descripción de la tabla**	
Opportunity	**Tabla utilizada para almacenar los datos de las Oportunidades**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
IsActive	Permite identificar si la oportunidad esta activa	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
ModifiedBy	Campo identificador del ultimo usuario que modifico el registro	Int
Name	Nombre de la Oportunidad	nvarchar
Ts	Almacena información relacionada a la zona horaria	datetime
LeadSourceId	**FK (LeadSource)**	Int
OpportunityStatusId	**FK (OpportunityStatus)**	Int
OwnerId	**FK (SmaUser)**	Int
ScaleTypeId	**FK (ScaleType)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 32 Diccionario de datos tabla Opportunity

Nombre de la tabla	**Descripción de la tabla**	
OpportunityContact	**tabla utilizada para la relación entre las tablas Opportunity y Contact**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la relación OpportunityContact	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsCustomer	Marca el contacto como customer	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
IsPrimary	Marca el contacto como el principal	Boolean
IsSalesLead	Marca el contacto como Sales lead	Boolean
IsSalesSupport	Define si el contacto es de soporte a ventas	Boolean
IsVendor	Define si el contacto es el vendedor	Boolean
Note	Permite agregar una nota a la creación del contacto de la oportunidad	nvarchar
Ts	Almacena información relacionada a la zona horaria	datetime
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
ContactId	**FK (Contact)**	Int

Nombre de la tabla	**Descripción de la tabla**	
OpportunityContact	**tabla utilizada para la relación entre las tablas Opportunity y Contact**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
OpportunityId	**FK (Opportunity)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 33 Diccionario de datos tabla OpportunityContact

Nombre de la tabla	**Descripción de la tabla**	
Person	**tabla utilizada para guardar datos de Personas**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la persona	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
FirstName	Almacena el primer nombre de la persona	nvarchar
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
JobTitle	Guarda la ocupación de la persona	nvarchar
LastName	Almacena el apellido de una persona	nvarchar
Mobile	Almacena el número telefónico de la persona	varchar
Ts	Almacena información relacionada a la zona horaria	datetime
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 34 Diccionario de datos tabla Person

Sprint #2

Base de datos del SMA

Nombre de la tabla	**Descripción de la tabla**	
SmaApplication	**Lista de aplicaciones que pueden hacer uso del servicio de autenticación**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la aplicación	Int
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
IsActive	Identifica si la aplicación esta activa	Boolean
Name	Nombre de la aplicación	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 35 Diccionario de datos tabla SmaApplication

Nombre de la tabla	**Descripción de la tabla**	
SmaClient	**Lista de clientes de aplicación que pueden hacer uso del servicio de autenticación**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del cliente para autenticación	Int
AllowedOrigin	Nombres de dominio desde donde son permitidas las solicitudes	nvarchar
ApplicationTypes	Tipo de aplicación	Int
Base64Secret	Hash identificador de la aplicación	nvarchar
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
IsActive	Identifica si el cliente del tenant está activo	Boolean
Name	Nombre del cliente	nvarchar
RefreshTokenLifeTime	Duración de la sesión en la aplicación	Int

Nombre de la tabla	**Descripción de la tabla**	
SmaClient	**Lista de clientes de aplicación que pueden hacer uso del servicio de autenticación**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Ts	Fecha en que se agregó el registro en segundos	timestamp
ApplicationId	**FK (SmaApplication)**	Int
ClientId	**FK (SmaClient)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 36 Diccionario de datos tabla SmaClient

Nombre de la tabla	**Descripción de la tabla**	
SmaTenant	**Lista de clientes o tenants del sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Tenant	Int
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
Email	Email de quien registro el tenant	nvarchar
IsActive	Identifica si el tenant está activo	Boolean
Name	Nombre del Tenant	nvarchar
Phone	Número telefónico	nvarchar
SubDomain	Subdominio para acceso al tenant	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 37 Diccionario de datos tabla SmaTenant

Nombre de la tabla	**Descripción de la tabla**	
SmaTenantClient	**Asocia clientes al tenant**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
IsActive	Identifica si el registro está activo	Boolean

Nombre de la tabla	**Descripción de la tabla**	
SmaTenantClient	**Asocia clientes al tenant**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Ts	Fecha en que se agregó el registro en segundos	timestamp
ClientId	**FK (SmaClient)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 38 Diccionario de datos tabla SmaTenantClient

Base de datos del CRM

Nombre de la tabla	**Descripción de la tabla**	
Address	**Tabla utilizada para almacenar las direcciones**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la dirección	Int
Address1	Primera parte de la dirección	nvarchar
Address2	Segunda parte de la dirección	nvarchar
City	Ciudad de la dirección	nvarchar
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
StateProvince	Nombre del estado o provincia de la dirección	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
ZipPostalCode	Numero de código postal	nvarchar
CountryId	**FK (Country)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
StateId	**FK (State)**	Int

Tabla 39 Diccionario de datos tabla Address

Nombre de la tabla	**Descripción de la tabla**	
AddressPurpose	**Almacena los propósitos de las direcciones**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del registro de AddressPurpose	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del propósito de la dirección	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 40 Diccionario de datos tabla AddressPurpose

Nombre de la tabla	**Descripción de la tabla**	
CityDMARegionCode	**DMA (Designated Market Area) Lista de ciudades para un DMA**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del DMA	Int
City	Nombre de la ciudad	nvarchar
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
DmaRegionCodeId	**FK (CityDMARegionCode)**	Int
ModifiedBy	**FK (SmaUser)**	Int
StateId	**FK (State)**	Int

Tabla 41 Diccionario de datos tabla CityDMARegionCode

Nombre de la tabla	**Descripción de la tabla**	
Country	**Tabla con la lista de países**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	**Identificador del país**	Int
Code	Código del país	nvarchar

Nombre de la tabla	**Descripción de la tabla**	
Country	**Tabla con la lista de países**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si el país está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del país	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	**Int**
ModifiedBy	**FK (SmaUser)**	**Int**

Tabla 42 Diccionario de datos tabla Country

Nombre de la tabla	**Descripción de la tabla**	
County	**Almacena la lista de condados**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del condado	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del condado	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
StateId	**FK (State)**	Int

Tabla 43 Diccionario de datos tabla County

Nombre de la tabla	**Descripción de la tabla**	
CustomerAddress	**Direcciones asociadas a los clientes del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del CustomerAddress	Int

Nombre de la tabla	**Descripción de la tabla**	
CustomerAddress	**Direcciones asociadas a los clientes del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
AddressId	**FK (Address)**	Int
AddressPurposeId	**FK (AddressPurpose)**	Int
CustomerId	**FK (Customer)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 44 Diccionario de datos tabla CustomerAddress

Nombre de la tabla	**Descripción de la tabla**	
DMARegionCode	**Direcciones asociadas a los clientes del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del DMA	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
DmaCode	Código del DMA	smallint
DmaRegion	Nombre de la región	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 45 Diccionario de datos tabla DMARegionCode

Nombre de la tabla	**Descripción de la tabla**	
File	**Tabla con la lista de archivos subidos al sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del archivo	Int

Nombre de la tabla	**Descripción de la tabla**	
File	**Tabla con la lista de archivos subidos al sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Extension	Identificador del tipo de archivo	nvarchar
FileName	Nombre del archivo	nvarchar
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Location	Ruta física del archivo	nvarchar
Size	Espacio utilizado en HDD	bigint
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OwnerId	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 46 Diccionario de datos tabla file

Nombre de la tabla	**Descripción de la tabla**	
FilePointer	**Tabla utilizada para asociar archivos a módulos del sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del registro	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
ModuleType	**FK (ModuleType)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
RegardingObjectId	**FK (File)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 47 Diccionario de datos tabla FilePointer

Nombre de la tabla	**Descripción de la tabla**	
OpportunityStage	**Almacena la asociación de las fases por los que pasa una Oportunidad**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	**Identificador del OpportunityStage**	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si la fase asociada esta activa	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OpportunityId	**FK (Opportunity)**	Int
StageId	**FK (Stage)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 48 Diccionario de datos tabla OpportunityStage

Nombre de la tabla	**Descripción de la tabla**	
Stage	**Lista de las fases o etapas por las que pasara una Oportunidad**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la etapa	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si la fase está activa	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del Stage	nvarchar
PercentProbability	Porcentaje de completado para la etapa	tinyint
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	FK (SmaUser)	Int
ModifiedBy	FK (SmaUser)	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 49 Diccionario de datos tabla Stage

Nombre de la tabla	**Descripción de la tabla**	
State	**Lista de los estados de un país**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del State	Int
Code	Código del estado del país	nvarchar
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del estado	nvarchar
StateType	Tipo de estado	tinyint
Ts	Fecha en que se agregó el registro en segundos	timestamp
CountryId	**FK (Country)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 50 Diccionario de datos tabla State

Nombre de la tabla	**Descripción de la tabla**	
ZipCode	**Lista de los ZipCodes**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del ZipCode	Int
City	Nombre de la ciudad a la que pertenece el ZipCode	nvarchar
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
ZipCodeValue	Numero de ZipCode	nvarchar
CountyId	**FK (County)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 51 Diccionario de datos tabla ZipCode

Sprint #3

Base de datos del CRM

Nombre de la tabla	**Descripción de la tabla**	
Lead	**Almacena la lista de Leads**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Lead	Int
Company	Nombre de la compañía del Lead	nvarchar
CustomerAccountType	Identifica la oportunidad como comercial o residencial	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
FollowUpDate	Fecha para dar seguimiento al Lead	datetime
IsActive	Identifica si el lead está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
LeadSourceId	**FK (LeadSource)**	Int
ModifiedBy	**FK (SmaUser)**	Int
PersonId	**FK (Person)**	Int
ScaleTypeId	**FK (ScaleType)**	Int
StatusLeadId	**FK (StatusLead)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 52 Diccionario de datos tabla Lead

Nombre de la tabla	**Descripción de la tabla**	
LeadSource	**Utilizada para almacenar las fuentes de los Leads**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del LeadSource	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2

Nombre de la tabla	**Descripción de la tabla**	
LeadSource	**Utilizada para almacenar las fuentes de los Leads**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del LeadSource	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 53 Diccionario de datos tabla LeadSource

Nombre de la tabla	**Descripción de la tabla**	
OpportunityStatus	**Lista de estados de las oportunidades**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del estado de la oportunidad	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del estado de la oportunidad	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 54 Diccionario de datos tabla OpportunityStatus

Nombre de la tabla	**Descripción de la tabla**	
RecurrentPatternSchedule	**Registro de eventos que se repiten en calendario**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	**Identificador del RecurrentPatternSchedule**	Int
DateCreated	Almacena la fecha de creación	datetime2

Nombre de la tabla	**Descripción de la tabla**	
RecurrentPatternSchedule	**Registro de eventos que se repiten en calendario**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateModified	Almacena la fecha de la última modificación	datetime2
Details	Detalles del evento	nvarchar
Duration	Duración del evento	Int
EndDate	Fecha de finalización	datetimeoffset
EndTypeRecurrence	Tipo de finalización del evento	Int
EveryItem	Número de veces que se repite el evento	Int
IsActive	Identifica si el evento recurrente está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
IsWholeDay	Determina si el evento es para el día completo	Boolean
Latitude	Campo de ubicación geográfica	numeric
Location	Nombre o dirección del evento	nvarchar
Longitude	Campo de ubicación geográfica	numeric
MaxOccurrences	Máximo de veces en que se repetirá el evento	Int
PatternDaily	Determina si el evento es para un día especifico a para todos los días de la semana	Int
ReccurencePattern	Días de la semana en que se repite el evento	varchar
RecurrenceDay	Día del mes en que se repite el evento	Int
StartDate	Fecha de inicio del evento	datetimeoffset
Subject	Nombre del evento en calendario	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
CustomerId	**FK (Customer)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OwnerId	**FK (SmaUser)**	Int
PatternId	**FK (Schedule)**	Int
ScheduleTypeId	**FK (ScheduleType)**	Int

Nombre de la tabla	**Descripción de la tabla**	
RecurrentPatternSchedule	**Registro de eventos que se repiten en calendario**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
TenantId	**FK (SmaTenant)**	Int

Tabla 55 Diccionario de datos tabla RecurrentPatternSchedule

Nombre de la tabla	**Descripción de la tabla**	
ScaleType	**Lista de etapas que define que tan cerca está el Lead en ser una Oportunidad**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del ScaleType	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del ScaleType	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 56 Diccionario de datos tabla ScaleType

Nombre de la tabla	**Descripción de la tabla**	
Schedule	**Tabla para las actividades agregadas en calendario**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del evento en calendario	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Details	Detalles sobre el evento	nvarchar
EndDate	Fecha de finalización del evento	datetimeoffset
IsActive	Identifica si el evento está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean

Nombre de la tabla	**Descripción de la tabla**	
Schedule	**Tabla para las actividades agregadas en calendario**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
IsWholeDay	Determina si el evento es para el día completo	Boolean
Latitude	Campo de ubicación geográfica	numeric
Location	Nombre o dirección del evento	nvarchar
Longitude	Campo de ubicación geográfica	numeric
StartDate	Fecha de inicio del evento	datetimeoffset
Subject	Nombre del evento	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
AddressId	**FK (Address)**	Int
CreatedBy	**FK (SmaUser)**	Int
CustomerId	**FK (Customer)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int
OwnerId	**FK (SmaUser)**	Int
ScheduleTypeId	**FK (ScheduleType)**	Int
RecurrentPatternScheduleId	**FK (RecurrentPatternSchedule)**	Int

Tabla 57 Diccionario de datos tabla Schedule

Nombre de la tabla	**Descripción de la tabla**	
ScheduleType	**Tipos de actividades en el calendario**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	**Identificador del tipo de evento**	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si el tipo de evento está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del tipo de actividad en calendario	varchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 58 Diccionario de datos tabla ScheduleType

Nombre de la tabla	**Descripción de la tabla**	
StatusLead	**Lista de estados para Lead**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del StatusLead	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del estado	varchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 59 Diccionario de datos tabla StatusLead

Nombre de la tabla	**Descripción de la tabla**	
Task	**Almacena la lista de tareas**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Task	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Details	Detalles de la tarea	nvarchar
Due	Fecha límite para terminación de la tarea	datetimeoffset
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre de la tarea	nvarchar
PercentComplete	Porcentaje de avance en la tarea	tinyint
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
CustomerId	**FK (Customer)**	Int
ModifiedBy	**FK (SmaUser)**	Int
OwnerId	**FK (SmaUser)**	Int
TaskStatusId	**FK (TaskStatus)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 60 Diccionario de datos tabla Task

Nombre de la tabla	**Descripción de la tabla**	
TaskStatus	**Lista de los estados de las tareas**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del estado de la tarea	Int

Nombre de la tabla	**Descripción de la tabla**	
TaskStatus	**Lista de los estados de las tareas**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del estado de la tarea	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 61 Diccionario de datos tabla TaskStatus

Sprint #4

Base de datos del SMA

Nombre de la tabla	**Descripción de la tabla**	
SmaAction	**Almacena la lista de Acciones del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Action	Int
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
IsActive	Identifica si el action está activo	Boolean
Name	Nombre del Action del sistema	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
ApplicationId	**FK (SmaApplication)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 62 Diccionario de datos tabla SmaAction

Nombre de la tabla	**Descripción de la tabla**	
SmaModule	**Almacena la lista de módulos**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del modulo	Int
DateCreated	Almacena la fecha de creación	datetime

Nombre de la tabla	**Descripción de la tabla**	
SmaModule	**Almacena la lista de módulos**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateModified	Almacena la fecha de la última modificación	datetime
Description	Descripción del modulo	nvarchar
IsActive	Identifica si el modulo está activo	Boolean
Name	Nombre del Modulo	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
ApplicationId	**FK (SmaApplication)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 63 Diccionario de datos tabla SmaModule

Nombre de la tabla	**Descripción de la tabla**	
SmaModuleAction	**Tabla utilizada para asociar una acción a su modulo**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
IsActive	Identifica si la asociación de modulo y acción esta activa	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
ActionId	**FK (SmaAction)**	Int
ApplicationId	**FK (SmaApplication)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
ModuleId	**FK (Module)**	Int

Tabla 64 Diccionario de datos tabla SmaModuleAction

Nombre de la tabla	**Descripción de la tabla**	
SmaRole	**Almacena la lista de Roles**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	**Identificador del Role**	Int
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime

Nombre de la tabla	**Descripción de la tabla**	
SmaRole	**Almacena la lista de Roles**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
IsSysAdmin	Identificado el Role como administrador del sistema	Boolean
Name	Nombre del role	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 65 Diccionario de datos tabla SmaRole

Nombre de la tabla	**Descripción de la tabla**	
SmaRoleAction	**Tabla utilizada para asociar una acción a un rol**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
IsActive	Identifica si la asociación de role y acción esta activa	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
ActionId	**FK (Action)**	Int
ApplicationId	**FK (SmaApplication)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
ModuleId	**FK (Module)**	Int
RoleId	**FK (Role)**	Int

Tabla 66 Diccionario de datos tabla SmaRoleAction

Nombre de la tabla	**Descripción de la tabla**	
SmaUserAction	**Almacena las acciones permitidas a un usuario del sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime

Nombre de la tabla	Descripción de la tabla	
SmaUserAction	**Almacena las acciones permitidas a un usuario del sistema**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
IsActive	Identifica si la asociación de actions a usuarios esta activa	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
ActionId	**FK (Action)**	Int
ApplicationId	**FK (SmaApplication)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
ModuleId	**FK (Module)**	Int
UserId	**FK (SmaUser)**	Int

Tabla 67 Diccionario de datos tabla SmaUserAction

Nombre de la tabla	Descripción de la tabla	
SmaUserRole	**Asociación de Role a un usuario del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime
DateModified	Almacena la fecha de la última modificación	datetime
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
RoleId	**FK (Role)**	Int
UserId	**FK (SmaUser)**	Int

Tabla 68 Diccionario de datos tabla SmaUserRole

Base de datos del CRM

Nombre de la tabla	Descripción de la tabla	
ActivityLog	**Tabla utilizada para un registro de las acciones realizadas en el CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Activity	Int

Nombre de la tabla	**Descripción de la tabla**	
ActivityLog	**Tabla utilizada para un registro de las acciones realizadas en el CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Detail	Detalle de la acción realizada	nvarchar
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
ActivityTypeId	**FK (ActivityType)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
ModuleTypeId	**FK (ModuleType)**	Int
OwnerId	**FK (SmaUser)**	Int
ParentActivityLogId	**FK (ActivityLog)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 69 Diccionario de datos tabla ActivityLog

Nombre de la tabla	**Descripción de la tabla**	
ActivityLogObject	**Asociación de las Actividades en el CRM con los distintos módulos**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del ActivityLogObject	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre asociado al Activity	nvarchar
ObjectId	Identificador del registro en el que se realizó alguna acción	Int
Ts	Fecha en que se agregó el registro en segundos	timestamp
ActivityLogId	**FK (ActivityLog)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 70 Diccionario de datos tabla ActivityLogObject

Nombre de la tabla	**Descripción de la tabla**	
ActivityType	**Tipos de actividades que se permiten realizar en el CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del tipo de actividad	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre de la actividad	varchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 71 Diccionario de datos tabla ActivityType

Nombre de la tabla	**Descripción de la tabla**	
CrmxSettings	**Almacena la configuración la empresa que hace uso del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del registro de configuración	Int
CultureInfoName	Código de idioma	nvarchar
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
OrganizationName	Nombre de la empresa que hace uso del CRM	nvarchar
StandardOffset	Valor de diferencia horaria (-06:00)	nvarchar
TimeZoneInfo	Nombre de la zona horaria por defecto	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CountryId	**FK (Country)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
StateId	**FK (State)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 72 Diccionario de datos tabla CrmxSettings

Nombre de la tabla	**Descripción de la tabla**	
CustomerCustomFieldValue	**Almacena los CustomFields de los customers**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del CustomField para un Customer	Int

Nombre de la tabla	**Descripción de la tabla**	
CustomerCustomFieldValue	**Almacena los CustomFields de los customers**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Datatype	Tipo de dato del CustomField	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Label	Nombre de descripción para el CustomField para Customer	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
Value	Valor del CustomField	nvarchar
CustomerId	**FK (Customer)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 73 Diccionario de datos tabla CustomerCustomFieldValue

Nombre de la tabla	**Descripción de la tabla**	
CustomField	**Almacena los campos personalizados utilizados en el CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del CustomField	Int
Datatype	Tipo de dato del CustomField	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
DefaultValue	Valor por defecto para el CustomField	nvarchar
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
IsRequired	Define si el CustomField es requerido	Boolean
Label	Nombre de descripción para el CustomField	nvarchar
SourceTable	Identificador de tabla o modulo al que pertenece el CustomField	Int
Ts	Fecha en que se agregó el registro en segundos	timestamp
Value	Valor insertado al CustomField	nvarchar
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Nombre de la tabla	**Descripción de la tabla**	
CustomField	**Almacena los campos personalizados utilizados en el CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
TenantId	**FK (SmaTenant)**	Int

Tabla 74 Diccionario de datos tabla CustomField

Nombre de la tabla	**Descripción de la tabla**	
CustomFieldValue	**Almacena los valores de los campos personalizados en el CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)		Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Label	Nombre de descripción para el CustomField	nvarchar
SourceTable	Nombre de la tabla a la que pertenece el CustomField	Int
Ts	Fecha en que se agregó el registro en segundos	timestamp
Value	Valor insertado al CustomField	nvarchar
CreatedBy	**FK (SmaUser)**	Int
Datatype	Tipo de dato del CustomField	Int
EntityId		Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 75 Diccionario de datos tabla CustomFieldValue

Nombre de la tabla	**Descripción de la tabla**	
Department	**Almacena la lista de Departamentos en una empresa**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Department	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si el Department está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del departamento	nvarchar

Nombre de la tabla	**Descripción de la tabla**	
Department	**Almacena la lista de Departamentos en una empresa**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
ParentId	**FK (Department)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 76 Diccionario de datos tabla Department

Nombre de la tabla	**Descripción de la tabla**	
Employee	**Employees o colaboradores de la empresa que hace uso del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	**Identificador del Employee**	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Email	e-mail del colaborador	nvarchar
EndDate	Fecha en que salió de la compañía	date
Ext	Numero de extensión telefónica	varchar
Fax	Numero de Fax	varchar
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Phone	Número de teléfono	varchar
StartDate	Fecha en que inicio en la compañía	date
Title	Cargo que ocupa en la compañía	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ManagerId	**FK (Employee)**	Int
ModifiedBy	**FK (SmaUser)**	Int
PersonId	**FK (Person)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 77 Diccionario de datos tabla Employee

Nombre de la tabla	**Descripción de la tabla**	
EmployeeSetting	**Configuración de sistema para el Employee**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la configuración asociada al Employee	Int
CommissionRate	Porcentaje de comisión para el colaborador	numeric
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
EmailMeDailyToDo	Enviar notificaciones de eventos del calendario	Boolean
IsActive	Identifica si la configuración del Employee está activa	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
IsRestrictedByTerritory	Si el colaborador está restringido a un territorio	Boolean
StandardOffset	Valor de diferencia horaria (-06:00)	nvarchar
TimeZoneInfo	Nombre de la zona horaria por defecto	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CountryId	**FK (Country)**	Int
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
StateId	**FK (State)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 78 Diccionario de datos tabla EmployeeSetting

Nombre de la tabla	**Descripción de la tabla**	
EmployeeSetting	**Asociación de un Employee a un territorio o zona de trabajo**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del registro EmployeeTerritory	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si la asociación de Employee territorio esa activa	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
IsTeamLeader	Employee asociado es líder de equipo	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
EmployeeId	**FK (Employee)**	Int

Nombre de la tabla	**Descripción de la tabla**	
EmployeeSetting	**Asociación de un Employee a un territorio o zona de trabajo**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int
TerritoryId	**FK (Territory)**	Int

Tabla 79 Diccionario de datos tabla EmployeeTerritory

Nombre de la tabla	**Descripción de la tabla**	
EmployeeWorkStation	**Tabla de relación entre Employee y su WorkStation**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del registro	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si la asociación Employee y Workstation esta activa	Boolean
IsBoss	Identifica si es Jefe	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
EmployeeId	**FK (Employee)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int
WorkStationId	**FK (WorkStation)**	Int

Tabla 80 Diccionario de datos tabla EmployeeWorkStation

Nombre de la tabla	**Descripción de la tabla**	
MarketSegment	**Lista de segmentos de mercados**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del MarketSegment	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
Description	Descripción del MarketSegment	nvarchar
IsActive	Identifica si el MarketSegment está activo	Boolean

Nombre de la tabla	**Descripción de la tabla**	
MarketSegment	**Lista de segmentos de mercados**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del MarketSegment	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 81 Diccionario de datos tabla MarketSegment

Nombre de la tabla	**Descripción de la tabla**	
ModuleType	**Lista de módulos del CRM**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del modulo	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre del modulo	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
ModifiedBy	**FK (SmaUser)**	Int

Tabla 82 Diccionario de datos tabla ModuleType

Nombre de la tabla	**Descripción de la tabla**	
StaffEmployee		
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)		Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive		Boolean
IsManager		Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
AspNetUsersId	**FK (SmaUser)**	Int

Nombre de la tabla	**Descripción de la tabla**	
StaffEmployee		
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
CreatedBy	**FK (SmaUser)**	Int
EmployeeId	**FK (Employee)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 83 Diccionario de datos tabla StaffEmployee

Nombre de la tabla	**Descripción de la tabla**	
Territory	**Lista de Territorios o zonas de trabajo**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador del Territory	Int
Abbreviation	Nombre corto del Territory	nvarchar
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si el territorio está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
LongName	Nombre completo del Territory	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
MarketSegmentId	**FK (MarketSegment)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 84 Diccionario de datos tabla Territory

Nombre de la tabla	**Descripción de la tabla**	
TerritoryZipCode	**ZipCodes asociados a un territorio**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	**Identificador del TerritoryZipCode**	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int

Nombre de la tabla	**Descripción de la tabla**	
TerritoryZipCode	**ZipCodes asociados a un territorio**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int
TerritoryId	**FK (Territory)**	Int
ZipCode	**FK (ZipCode)**	Int

Tabla 85 Diccionario de datos tabla TerritoryZipCode

Nombre de la tabla	**Descripción de la tabla**	
WorkStation	**Almacena la lista de estación de trabajo**	
Nombre de la columna	**Descripción de la columna**	**Tipo de datos**
Id (PK)	Identificador de la estación de trabajo	Int
DateCreated	Almacena la fecha de creación	datetime2
DateModified	Almacena la fecha de la última modificación	datetime2
IsActive	Identifica si el workStation está activo	Boolean
IsDeleted	Campo utilizado para borrado lógico del registro	Boolean
Name	Nombre de la estación de trabajo	nvarchar
Ts	Fecha en que se agregó el registro en segundos	timestamp
CreatedBy	**FK (SmaUser)**	Int
DepartmentId	**FK (Department)**	Int
ModifiedBy	**FK (SmaUser)**	Int
TenantId	**FK (SmaTenant)**	Int

Tabla 86 Diccionario de datos tabla WorkStation

Web Services

URL	Parámetro esperado	Tipo	Descripción
api/territories/bulkAction/delete		Post	En Territorio, Eliminar múltiples registros.
api/territories		Post	En Territorio, Almacenar registro.
api/territories/{id:int}	id:int	Delete	En Territorio, Eliminar un registro.
api/territories/list		Get	En Territorio, Obtener lista de todos los registros.
api/territories/{id:int}	id:int	Get	En Territorio, Obtener un registro.
api/territories/names		Get	En Territorio, Obtener los nombres de registros.
api/territories/{id:int}	id:int	Put	En Territorio, Actualizar un registro.
api/files/bulkAction/delete		Post	En Archivo, Eliminar múltiples registros.
api/files/		Post	En Archivo, Almacenar registro.
api/files/{id:int}	id:int	Delete	En Archivo, Eliminar un registro.
api/files/{id:int}	id:int	Get	En Archivo Obtener un registro.
api/files/list		Get	En Archivo, Obtener lista de todos los registros.
api/contacts/bulkAction/delete		Post	En Contacto, Eliminar múltiples registros.
api/contacts/bulkAction/reAssignOwner		Post	En Contacto, Reasignación múltiple de propietario en registro.
api/contacts/		Post	En Contacto, Almacenar un registro.
api/contacts/{id:int}	id:int	Delete	En Contacto, Eliminar un registro.
api/contacts/{id:int}	id:int	Get	En Contacto, Obtener un registro.
api/contacts/{id:int}/accounts/list	id:int	Get	En Contacto, Obtener lista de todos los registros de un contacto.
api/contacts/{id:int}/detail	id:int	Get	En Contacto, Obtener detalles de un contacto.
api/contacts/list		Get	En Contacto, Obtener lista de todos los contactos.
api/contacts/{id:int}/opportunities/list	id:int	Get	En Contacto, Obtener lista de todas las oportunidades de un contacto.

URL	Parámetro esperado	Tipo	Descripción
api/contacts/{id:int}	id:int	Put	En Contacto, Actualizar un contacto.
api/accountContacts/		Post	En Contactos de cuenta, Enlazar un contacto a una cuenta.
api/accountContacts/{id:int}	id:int	Delete	En Contactos de cuenta, Retirar un contacto de una cuenta
api/accounts/{id:int}/accountContacts/list	id:int	Get	En Contactos de cuenta, Obtener lista de todos los registros enlazados.
api/accountContacts/{id:int}	id:int	Get	En Contactos de cuenta, Obtener un registro.
api/accountContacts/{id:int}	id:int	Put	En Contactos de cuenta, Actualizar un registro.
api/accounts/bulkAction/delete		Post	En Cuentas, Eliminar múltiples cuentas.
api/accounts/bulkAction/reAssignOwner		Post	En Cuentas, Reasignación múltiple de propietario.
api/accounts		Post	En Cuentas, Almacenar un registro.
api/accounts/{id:int}	id:int	Delete	En Cuentas, Eliminar un registro.
api/accounts/{id:int}/downgrade	id:int	Post	En Cuentas, degradar una cuenta a oportunidad.
api/accounts/{id:int}	id:int	Get	En Cuentas, Obtener una cuenta.
api/accounts/{id:int}/detail	id:int	Get	En Cuentas, Obtener detalles de una cuenta.
api/accounts/followUps		Get	En Cuentas, mostrar datos de seguimiento.
api/accounts/names/list/{customerAccountType:int?}	customerAccountType:int?	Get	En Cuentas, Obtener lista de un tipo de cuentas específico.
api/accounts/list		Get	En Cuentas, Obtener lista de todas las cuentas.
api/accounts/{id:int}/opportunity/list	id:int	Get	En Cuentas, Obtener lista de todas las oportunidades de una cuenta.
api/accounts/{id:int}	id:int	Put	En Cuentas, Actualizar una cuenta.
api/schedules/recurrent		Post	En Calendario Almacenar registro recurrente.

URL	Parámetro esperado	Tipo	Descripción
api/schedules		Post	En Calendario un registro.
api/schedules/recurrent/{id:int}	id:int	Delete	En Calendario Eliminar un registro recurrente.
api/schedules/{id:int}	id:int	Delete	En Calendario Eliminar un registro.
api/schedules/today		Get	En Calendario obtener la información del día.
api/schedules/recurrent/{id:int}	id:int	Get	En Calendario Obtener un registro recurrente.
api/schedules/{id:int}	id:int	Get	En Calendario Obtener un registro.
api/schedules/list		Get	En Calendario Obtener lista de todos los registros.
api/schedules/recurrent/{id:int}	id:int	Put	En Calendario Actualizar un registro recurrente.
api/schedules/{id:int}	id:int	Put	En Calendario Actualizar un registro.
api/employees/{employeeId:int}/customerNotes	employeeId:int	Post	En Empleados, Almacenar una nota en un registro empleado.
api/employees/{employeeId:int}/customerNotes/list	employeeId:int	Get	En Empleados, Obtener lista de las notas del cliente.
api/leads/bulkAction/upgradeToOpportunity		Post	En Leads, Elevar múltiples Leads a oportunidades.
api/leads/bulkAction/delete		Post	En Leads, Eliminar múltiples leads.
api/leads/bulkAction/reassignOwner		Post	En Leads, Reasignación de propietario a múltiples leads.
api/leads		Post	En Leads Almacenar un registro.
api/leads/{id:int}	id:int	Delete	En Leads, Eliminar un registro.
api/leads/{id:int}	id:int	Get	En Leads, Obtener un registro.
api/leads/{id:int}/detail	id:int	Get	En Leads Obtener detalles de un registro.
api/leads/followUps		Get	En Leads, Obtener lista de seguimiento.
api/leads/list		Get	En Leads, Obtener lista de todos los registros.
api/leads/{id:int}	id:int	Put	En Leads, Actualizar un registro.

URL	Parámetro esperado	Tipo	Descripción
api/leads/{id:int}/upgradeToOpportunity	id:int	Post	En Leads, elevar un registro a Oportunidad.
api/leads/{key:guid}	key:guid	Post	En Leads, Almacenar un registro.
api/statusleads		Post	En Estado del lead, Almacenar un registro.
api/statusleads/{ id:int}	id:int	Delete	En Estado del lead, Eliminar un registro.
api/statusleads/list		Get	En Estado del lead, Obtener lista de todos los registros.
api/statusleads/{id:int}	id:int	Get	En Estado del lead, Obtener un registro.
api/statusleads/{id:int}	id:int	Put	En Estado del lead, Actualizar un registro.
api/opportunityContacts		Post	En Contactos de oportunidad, Almacenar registro.
api/opportunityContacts/{id:int}	id:int	Delete	En Contactos de oportunidad, Eliminar un registro.
api/opportunityContacts/{id:int}	id:int	Get	En Contactos de oportunidad, Obtener un registro.
api/opportunities/{id:int}/opportunityContacts/list	id:int	Get	En Oportunidades, Obtener lista de todos los Contactos para una oportunidad.
api/opportunityContacts/{id:int}	id:int	Put	En Contactos de oportunidad, Actualizar un registro.
api/opportunities/bulkAction/downgrade		Post	En Oportunidades, degradar múltiples registro a Leads.
api/opportunities/bulkAction/upgrade		Post	En Oportunidades, elevar múltiples registros a Cuentas.
api/opportunities/bulkAction/delete		Post	En Oportunidades, Eliminar múltiples registro.
api/opportunities/bulkAction/reAssignOwner		Post	En Oportunidades, Reasignar propietario a múltiples registros.
api/opportunities/{id:int}/changeStage/{stageId:int}	id:int	Put	En Oportunidades, cambiar el estado de un registro.

URL	Parámetro esperado	Tipo	Descripción
api/opportunities		Post	En Oportunidades, almacenar registro.
api/opportunities/{id:int}	id:int	Delete	En Oportunidades, Eliminar un registro.
api/opportunities/{id:int}/downgrade	id:int	Post	En Oportunidades, degradar un registro a Lead.
api/opportunities/inProgressList		Get	En Oportunidades, obtener la lista de estado "en progreso".
api/opportunities/lostList		Get	En Oportunidades, obtener lista de estado "Perdido".
api/opportunities/wonList		Get	En Oportunidades, obtener lista de estado "Ganado".
api/opportunities/{id:int}	id:int	Get	En Oportunidades, Obtener un registro.
api/opportunities/{id:int}/detail	id:int	Get	En Oportunidades Obtener detalles de un registro.
api/opportunities/followUps		Get	En Oportunidades, Obtener lista de seguimiento.
api/opportunities/list		Get	En Oportunidades, Obtener lista de todos los registros.
api/opportunities/account/{accountId:int}/list	accountId:int	Get	En Oportunidades, Obtener lista de todos los registros de una cuenta.
api/opportunities/{id:int}/lost	id:int	Put	En Oportunidades, Actualizar un registro a "Perdido"
api/opportunities/{id:int}/won	id:int	Put	En Oportunidades, Actualizar un registro a "Ganado"
api/opportunities/{id:int}	id:int	Put	En Oportunidades, Actualizar un registro.
api/opportunitystage		Post	En Estado de oportunidad, Agregar un registro.
api/opportunitystage/{id:int}	id:int	Delete	En Estados de oportunidad, Eliminar un registro.
api/opportunitystage/list		Get	En Estados de oportunidad, Obtener lista de todos los registros.

URL	Parámetro esperado	Tipo	Descripción
api/opportunitystage/{id:int}	id:int	Get	En Estados de oportunidad, Obtener un registro.
api/opportunitystage/{id:int}	id:int	Put	En Estados de oportunidad, Actualizar un registro.
api/customfields/bulkAction/delete		Post	En Campos personalizados, Eliminar múltiples registro.
api/customfields		Post	En Campos personalizados, Almacenar registro.
api/customfields/{id:int}	id:int	Delete	En Campos personalizados, Eliminar un registro.
api/customfields/list		Get	En Campos personalizados, Obtener lista de todos los registros.
api/customfields/{id:int}	id:int	Get	En Campos personalizados, Obtener un registro.
api/customfields/{id:int}	id:int	Put	En Campos personalizados, Actualizar un registro.
api/customFieldValue		Post	En Campos personalizados, Agregar valor a un registro.
api/customFieldValue/{id:int}	id:int	Delete	En Campos personalizados, Retirar valor de un registro.
api/customFieldValue/list		Get	En Campos personalizados, Obtener lista de valores de un registro.
api/customFieldValue/{id:int}	id:int	Get	En Campos personalizados, Obtener valor de un registro.
api/customFieldValue/{id:int}	id:int	Put	En Campos personalizados, Actualizar valor de un registro.
api/settings		Put	En Ajustes, Actualizar registros
api/settings		Get	En Ajustes, Obtener registros.
api/userwidgets		Post	En Widget, Almacenar selección del usuario.
api/userwidgets		Get	En Widget, Obtener selección del usuario.
api/widgets		Get	En Widget, Obtener listado de registros.

URL	Parámetro esperado	Tipo	Descripción
api/employees/{id:int}/territories	id:int	Post	En Empleados, Asignar un territorio a un registro.
api/employees		Post	En Empleados, Almacenar registro.
api/employees/{id:int}	id:int	Delete	En Empleados, Eliminar un registro.
api/employees/bulkAction/delete		Post	En Empleados, Eliminar múltiples registros.
api/employees/{id:int}/accounts/list	id:int	Get	En Empleados, Obtener lista de cuentas enlazadas al registro.
api/employees/{id:int}	id:int	Get	En Empleados, Obtener un registro.
api/employees/{id:int}/detail	id:int	Get	En Empleados, Obtener detalles de un registro.
api/employees/managers/names		Get	En Empleados, Obtener nombres de managers.
api/employees/list		Get	En Empleados, Obtener lista de todos los registros
api/employees/{id:int}/opportunity/list	id:int	Get	En Empleados, Obtener lista de oportunidades de un registro.
api/employees/{id:int}/schedules/list	id:int	Get	En Empleados, Obtener lista de calendario de un registro.
api/employees/{id:int}/tasks/list	id:int	Get	En Empleados, Obtener lista de tareas de un registro.
api/employees/{id:int}/territories/list	id:int	Get	En Empleados, Obtener lista de territorios de un registro.
api/employees/{id:int}	id:int	Put	En Empleados, Actualizar un registro.
api/employeeTerritories/{id:int}	id:int	Delete	En Territorios de empleado, Eliminar un registro.
api/employeeTerritories/{id:int}	id:int	Put	En Territorio del empleado, Actualizar un registro.
api/employeeTerritories/		Post	En Territorio del empleado, Agregar un registro.
api/users/		Post	En Usuarios, Almacenar registro.
api/users/{id:int}	id:int	Delete	En Usuarios, Eliminar un registro.
api/users/list		Get	En Usuarios, Obtener lista de todos

URL	Parámetro esperado	Tipo	Descripción
			los registros.
api/users/settings		Put	En Usuarios, Actualizar ajustes de un registro.
api/users/{id:int}	id:int	Put	En Usuarios Actualizar, un registro.
api/departments		Post	En Departamentos, Almacenar registro.
api/departments/{id:int}	id:int	Delete	En Departamentos Eliminar un registro.
api/departments/{id:int}/workStation	id:int	Get	En Departamentos, Obtener lista de Estaciones de trabajo de un registro.
api/departments		Get	En Departamentos, Obtener registros.
api/departments/{id:int}	id:int	Put	En Departamentos, Actualizar un registro.
api/workStations		Post	En Estación de trabajo, Almacenar registro.
api/workStations/{id:int}	id:int	Delete	En Estación de trabajo, Eliminar un registro.
api/workStations/list		Get	En Estación de trabajo, Obtener lista de todos los registros.
api/workStations/{id:int}	id:int	Put	En Estación de trabajo, Actualizar un registro.
api/reports/accountCustomDetail		Get	En Reportes, Obtener datos personalizados de cuentas.
api/reports/accountDetail		Get	En Reportes, Obtener detalle de cuentas.
api/reports/accountSummary		Get	En Reportes, Obtener listado de cuentas.
api/reports/opportunityCustomDetail/columns		Get	En Reportes, Obtener detalle personalizado de oportunidades por columna.
api/reports/opportunityCustomDetail		Get	En Reportes, Obtener detalle de oportunidades personalizado.
api/reports/opportunityDet		Get	En Reportes, Obtener detalle de

URL	Parámetro esperado	Tipo	Descripción
ailByAccount			oportunidades por cuentas.
api/reports/opportunityDetailReport		Get	En Reportes, Obtener detalles de oportunidades.
api/reports/opportunitySummary		Get	En Reportes, Obtener listado de oportunidades.
api/reports/opportunitySummaryByAccountDetail		Get	En Reportes, Obtener listado de oportunidades por cuentas.
api/reports/leadCustomDetail/columns		Get	En Reportes, Obtener detalle de Leads por columnas.
api/reports/leadCustomDetail		Get	En Reportes, Obtener detalles personalizados de Leads.
api/reports/leadDetail		Get	En Reportes, Obtener detalles de Leads.
api/reports/leadSummary		Get	En Reportes, Obtener listado de Leads.
api/reports/taskSummary		Get	En Reportes, Obtener listado de tareas.
api/reports/contactDetail		Get	En Reportes, Obtener detalle de contactos.
api/reports/contactCustomDetail		Get	En Reportes, Obtener detalle personalizado de contactos.
api/reports/contactSummary		Get	En Reportes, Obtener listado de contactos.
api/leadSources/bulkAction/delete		Post	En Origen de Leads, Eliminar múltiples registros.
api/leadSources		Post	En Origen de Leads, Almacenar registro.
api/leadSources/{id:int}	id:int	Delete	En Origen de Leads, Eliminar un registro.
api/leadSources/list		Get	En Origen de Leads, Obtener lista de todos los registros.
api/leadSources/{id:int}	id:int	Get	En Origen de Leads, Obtener un registro.
api/leadSources/names		Get	En Origen de Leads, Obtener

URL	Parámetro esperado	Tipo	Descripción
			nombres de registros.
api/leadSources/{id:int}	id:int	Put	En Origen de Leads, Actualizar un registro.
api/marketSegments/bulkAction/delete		Post	En Segmento de mercado, Eliminar múltiples registros.
api/marketSegments		Post	En Segmento de mercado, Almacenar un registro.
api/marketSegments/{id:int}	id:int	Delete	En Segmento de mercado, Eliminar un registro.
api/marketSegments/{id:int}	id:int	Get	En Segmento de mercado, Obtener un registro.
api/marketSegments/names		Get	En Segmento de mercado, Obtener nombres de registros.
api/marketSegments/list		Get	En Segmento de mercado, Obtener lista de todos los registros.
api/marketSegments/{id:int}	id:int	Put	En Segmento de mercado, Actualizar un registro.
api/stages/bulkAction/delete		Post	En Etapas, Eliminar múltiples registros.
api/stages/		Post	En Etapas, Almacenar un registro.
api/stages/{id:int}	id:int	Delete	En Etapas, Eliminar un registro.
api/stages/list		Get	En Etapas, Obtener lista de todos los registros.
api/stages/{id:int}	id:int	Get	En Etapas, Obtener un registro.
api/stages/names		Get	En Etapas, Obtener lista de nombre de todos los registros.
api/stages/{id:int}	id:int	Put	En Etapas, Actualizar un registro.
api/tasks/bulk/delete		Post	En Tareas, Eliminar múltiples registros.
api/tasks/		Post	En Tareas, Almacenar un registro.
api/tasks/{id:int}	id:int	Delete	En Tareas, Eliminar un registro.
api/tasks/{id:int}	id:int	Get	En Tareas, Obtener un registro.
api/tasks/list		Get	En Tareas, Obtener lista de todos los registros.

URL	Parámetro esperado	Tipo	Descripción
api/tasks/reminders		Post	En Tareas, Agregar un recordatorio.
api/tasks/customer/{customerId:int}/list	customerId:int	Get	En Tareas, Obtener lista de todos los registros para un cliente.
api/tasks/{id:int}/completed	id:int	Put	En Tareas, Actualizar ha completado un registro.
api/tasks/{id:int}	id:int	Put	En Tareas, Actualizar un registro.
api/users/current		Get	En Usuarios actual, Obtener detalles del usuario.
api/users/current/profile		Get	En Usuarios, Obtener perfil del usuario.
api/users/current/profile		Put	En Usuarios, Actualizar perfil del registro.
api/users/{userId:int}/profile	userId:int	Put	En Usuarios, Actualizar perfil un registro
api/users/{id:int}	id:int	Put	En Usuarios, Actualizar un registro.

Tabla 87 Listado de Web Services del sistema

Manual de Usuario

Br. Jarbit Lira Br. Yasser López

Guía de implantación

Br. Jarbit Lira Br. Yasser López